

高等学校“十二五”应用型特色规划教材

C 语言实验、课程设计及网上训练教程

赵 欣 杜忠友 解艳艳 刘 浩 编著

姜玉波 姜庆娜 靳天飞 孙晓燕 张海林 李 锋 参编

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书按照教育部围绕计算思维进行计算机课程教学改革的思想编写,浓缩了作者多年来的教学实践及软件开发经验。本书内容先进,体系合理,包含大量紧密联系社会生活实际的精选范例程序。

本书共 5 章,内容包括 C 语言集成开发环境简介、C 语言程序设计实验、C 语言课程设计、VIPExam 考试学习资源数据库系统的应用、C 程序的常见错误及调试方法与技巧。本书每个实验项目都遵从读者的认知规律编写,使读者通过同步自测习题夯实编程基础,通过范例程序学习,借鉴编程思想、方法,通过实验项目丰富实践经验和提高编程及调试能力。此外,本书提供了具有自动批改、计分功能的同步自测习题的电子文档可免费下载。

本书可作为高校 C 语言实验及课程设计教材,也可作为继续教育、远程教育、社会培训的 C 语言实验教材,还可作为计算机爱好者的自学教材或参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目 (CIP) 数据

C 语言实验、课程设计及网上训练教程 / 赵欣等编著. —北京: 电子工业出版社, 2016.8

ISBN 978-7-121-28753-4

I. ①C… II. ①赵… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2016) 第 096446 号

策划编辑: 任欢欢

责任编辑: 任欢欢

印 刷:

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1 092 1/16 印张: 12.25 字数: 313.6 千字

版 次: 2016 年 8 月第 1 版

印 次: 2016 年 8 月第 1 次印刷

定 价: 30.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: 192910558 (QQ 群)。

前 言

C 语言是在国内外广泛使用的一种计算机语言。许多高校开设了 C 语言程序设计课程，作为一门实践性很强的课程，不仅要求学生学习基本概念、方法、语法规则，更重要的是进行大量的上机实践，只有通过大量实践体验、积累编程经验，才能真正提高编程和调试程序的能力。为此，作者结合多年来的教学实践、软件开发经验和体会，编写此教材作为与课程配套的实验及课程设计教程，以指导学生上机实践。

教材内容及特色

本教材针对学生的每一个编程实验任务都提供了一个可供模仿和借鉴的范例程序。每个范例程序都给出了详细注释。

教育家杜威认为教育即生活，教育是一种社会生活过程。本教材秉承这种教育理念，教材中的范例程序不是仅局限在解决抽象的数学问题上，而是涉及社会生活的各个方面。

本教材旨在充分利用信息技术解决教师批改书面作业的情况不能及时反馈给学生的难题。针对 C 语言每一章的教学内容都设计了同步自测习题，每章习题存放在一个具有自动批改、计分功能的电子文档中。每个文件含有密码，学生在计算机上做完提交后，教师告知密码，学生即可看到隐藏在文件里的标准答案和分数，便于学生及时改错。

本教材旨在充分利用网络技术和资源，开阔学生视野。许多高校图书馆都购买了 VIPExam 考试学习资源数据库，它收录了 1500 余个考试科目的数万套历年真题试卷和模拟试卷，其中包括 C 语言二级考试笔试和机试套题及答案。本教材第 4 章详细介绍了 VIPExam 考试学习资源数据库系统的功能及操作。编写此章内容得到了中科软股教育科技（北京）股份有限公司谢峰、李秋实老师的大力支持与帮助，在此表示衷心感谢。

本教材共 5 章。可作为高校 C 语言实验及课程设计教材，也可作为继续教育、远程教育、社会培训的 C 语言实验教材，还可作为计算机爱好者的自学教材或参考书。

第 1 章是 C 语言集成开发环境简介。详细地介绍了在 Visual C++ 6.0 和 Turbo C 3.0 环境下编写、编译、链接和运行一个 C 程序的全过程。

第 2 章是 C 语言程序设计实验。针对 C 语言课堂教学每一章的内容，在此给出了同步自测习题、范例程序解析、实验项目。通过每章实验的同步自测（可自动批改）帮助学生夯实基础，通过范例程序解析使学生学习算法构成、编程思想、方法和技巧，通过实验项目使学生提高编程和调试程序的能力。

第 3 章是 C 语言课程设计。详细介绍了作为范例项目的结构设计、详细设计、编码设计，对学生的课程设计项目提出了具体设计要求。

第 4 章是 VIPExam 考试学习资源数据库系统的应用。重点介绍了 C 语言二级考试专辑的功能及操作，详细介绍了 VIPExam 考试学习资源数据库系统的高级应用。

第 5 章是 C 程序的常见错误及调试方法与技巧。对大量 C 程序的常见错误进行分析，运用软件工程课程中的软件测试和软件调试方法，结合具体出错程序实例详细介绍了调试程序的方法。同时还介绍了使用 Visual C++ 6.0 调试工具及时发现程序错误的技巧。

教材相关资料下载

本教材中 C 语言的各章同步自测习题文件及使用方法的视频文件，可登录华信教育资源网<http://www.hxedu.com.cn>进行免费注册下载。

本教材由山东建筑大学承担 C 语言程序设计课程教学的多位一线教师编著。赵欣参与了第 2 章、第 3 章、第 4 章、第 5 章的编写工作及负责本书的统稿工作，靳天飞、李锋参与了第 1 章的编写工作，杜忠友、解艳艳、刘浩参与了第 2 章的编写工作，姜玉波、姜庆娜、孙晓燕、张海林参与了第 3 章的编写工作。

本教材在编写过程中难免有疏漏之处，恳请广大读者提出宝贵意见。作者电子邮箱：zxcopy@126.com。

赵 欣

2016 年 5 月

目 录

第 1 章	C 语言集成开发环境简介	1
1.1	Visual C++ 6.0 集成开发环境	1
1.1.1	启动 Visual C++ 6.0	1
1.1.2	C 程序的创建、编译、链接、运行	2
1.2	Turbo C++ 3.0 集成开发环境	4
1.2.1	启动 Turbo C++ 3.0	4
1.2.2	C 程序的创建、编译、链接、运行	5
第 2 章	C 语言程序设计实验	7
2.1	熟悉 C 语言程序集成开发环境	7
2.1.1	同步自测	7
2.1.2	范例程序解析	8
2.1.3	实验项目	9
2.2	数据类型、运算符、表达式	11
2.2.1	同步自测	11
2.2.2	范例程序解析	12
2.2.3	实验项目	14
2.3	顺序结构程序设计	16
2.3.1	同步自测	16
2.3.2	范例程序解析	18
2.3.3	实验项目	20
2.4	选择结构程序设计	22
2.4.1	同步自测	22
2.4.2	范例程序解析	26
2.4.3	实验项目	32
2.5	循环结构程序设计	33
2.5.1	同步自测	33
2.5.2	范例程序解析	37
2.5.3	实验项目	44
2.6	一维数组	46
2.6.1	同步自测	46
2.6.2	范例程序解析	49
2.6.3	实验项目	55

2.7	二维数组	56
2.7.1	同步自测	56
2.7.2	范例程序解析	60
2.7.3	实验项目	66
2.8	函数和编译预处理	67
2.8.1	同步自测	67
2.8.2	范例程序解析	71
2.8.3	实验项目	80
2.9	指针应用	81
2.9.1	同步自测	81
2.9.2	范例程序解析	85
2.9.3	实验项目	95
2.10	结构体应用	96
2.10.1	同步自测	96
2.10.2	范例程序解析	101
2.10.3	实验项目	107
2.11	文件应用	108
2.11.1	同步自测	108
2.11.2	范例程序解析	112
2.11.3	实验项目	119
第3章	C 语言课程设计	120
3.1	学生成绩管理系统	120
3.1.1	软件设计目的	120
3.1.2	软件结构设计	120
3.1.3	软件详细设计	121
3.1.4	软件编码设计	123
3.2	小学生算术运算试卷自动生成系统	135
3.2.1	软件设计目的	135
3.2.2	软件结构设计	135
3.2.3	软件详细设计	135
3.2.4	软件编码设计	136
3.3	课程设计项目	141
3.3.1	家庭药品管理系统	141
3.3.2	工资管理系统	142
3.3.3	小学生算术运算题自动批改系统	144
第4章	VIPEXam 考试学习资源数据库系统的应用	146
4.1	VIPEXam 考试学习资源数据库简介	146
4.2	C 语言二级考试笔试和机试考试数据库系统的应用	146

4.3	VIPEXam 考试学习资源数据库系统的高级应用	149
第 5 章	C 程序的常见错误及调试方法与技巧	156
5.1	C 程序的常见错误分析	156
5.2	C 程序的调试方法	176
5.3	C 程序的调试技巧	182
附录 A	C 语言网络资源分布简介	185
附录 B	实验报告格式要求	186
参考文献	187

第1章

C 语言集成开发环境简介

1.1 Visual C++ 6.0 集成开发环境

Visual C++是 Microsoft 公司 Visual Studio 程序开发软件包中的软件之一。Visual C++是可视化的集成开发环境,它主要面向 C++语言开发,同时兼容 C 语言。这里主要介绍 Visual C++ 6.0 中文企业版的使用方法。

1.1.1 启动 Visual C++ 6.0

Visual C++ 6.0 可随 Visual Studio 程序开发软件包一起安装或单独安装。因此它的启动方法也不同。

方法一:单击 Windows 桌面上的“开始”菜单→“程序”→选择“Microsoft Visual Studio 6.0”→选择“Microsoft Visual C++ 6.0”运行,如图 1-1 所示。

方法二:单击 Windows 桌面上的“开始”菜单→“程序”→选择“Microsoft Visual C++ 6.0”运行即可。

方法三:双击 Windows 桌面上的 Visual C++ 6.0 图标运行即可。

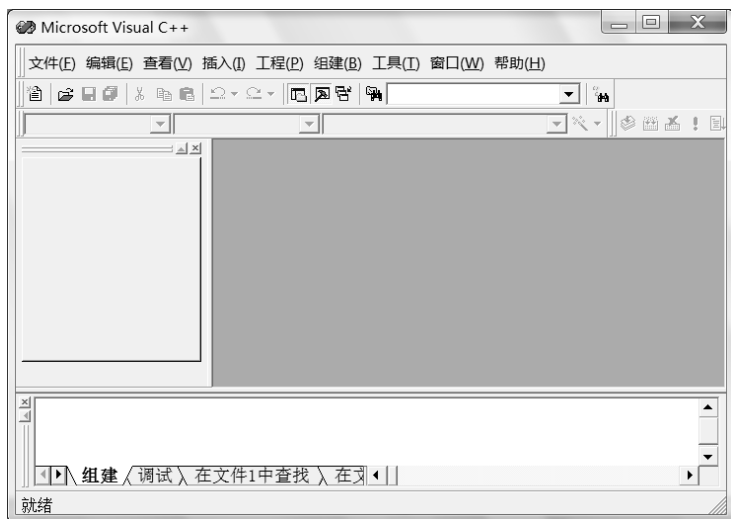


图 1-1 Visual C++ 6.0 集成开发环境窗口

1.1.2 C 程序的创建、编译、链接、运行

1. 建立一个新工程文件

首先在 D 盘上建立一个文件夹，以便于存放将要产生的一系列文件。例如文件夹 D:\PRG1032 (PRG 后跟学号的末 4 位)。

(1) 在 Visual C++中单击“文件”菜单→选择“新建”命令→在弹出的对话框中选择“工程”标签，如图 1-2 所示。

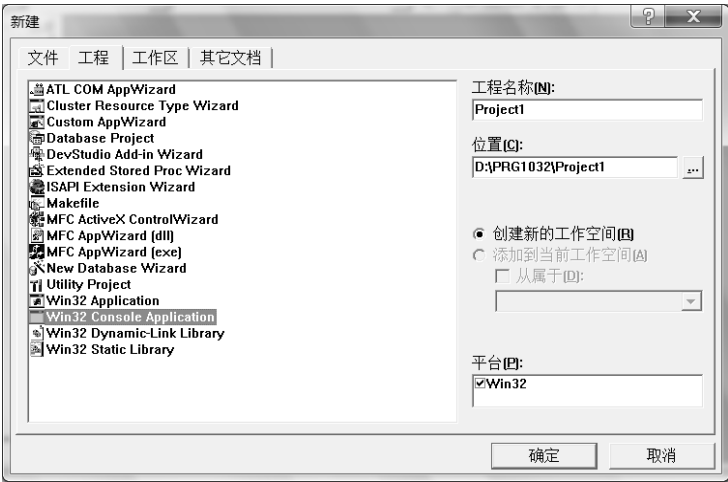


图 1-2 建立工程文件对话框

(2) 单击“Win32 Console Application”选择工程类型，在右边的“平台”选框中就会出现 Win32。

(3) 在右边的“位置”文本框中选择存放新工程文件的文件夹，例如，D:\PRG1032→在右边的“工程名称”文本框中输入工程文件名，例如 Project1→单击“确定”按钮。

(4) 在弹出的对话框中选择“一个空工程”→单击“完成”按钮→在新弹出的对话框中单击“确定”按钮，就建立了一个新工程，如图 1-3 所示。



图 1-3 选择建立新工程对话框

2. 创建 C 语言源程序文件

单击“文件”菜单→选择“新建”命令→在弹出的对话框中选择“文件”标签→选择“C++ Source File”→勾选“添加工程”复选框，激活其下面的选项→在文件名文本框输入 C 语言源程序文件名，例如，PRG1.C →单击“确定”按钮，如图 1-4 所示。

这时将出现源程序编辑窗口，可以在其中输入、编辑源程序。

注意：输入 C 语言源程序文件名时，必须指定扩展名.c，否则系统会自动以.cpp 为扩展名。

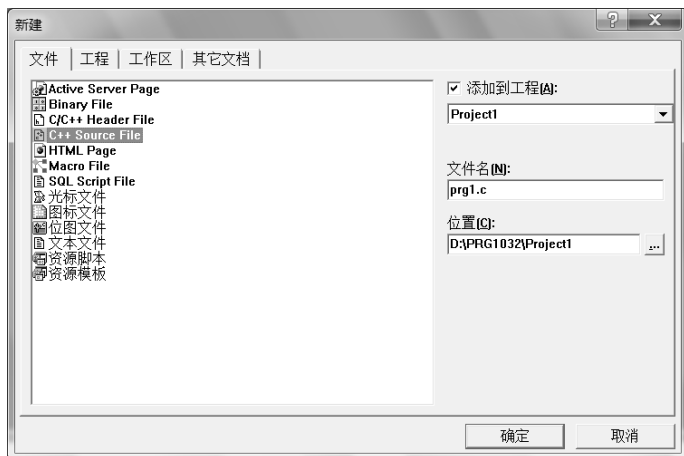



图 1-4 建立源程序文件窗口

3. 编译、链接和运行程序

(1) 编译产生.obj 文件。单击工具栏上的编译按钮，若编译成功则在下方的“组建”窗口将显示：0 error(s), 0 warning(s)。若程序有错误则显示错误提示信息，双击某行的错误提示，系统会自动显示一个箭头指向源程序的出错语句。继续修改、编译直到没有错误为止，如图 1-5 所示。

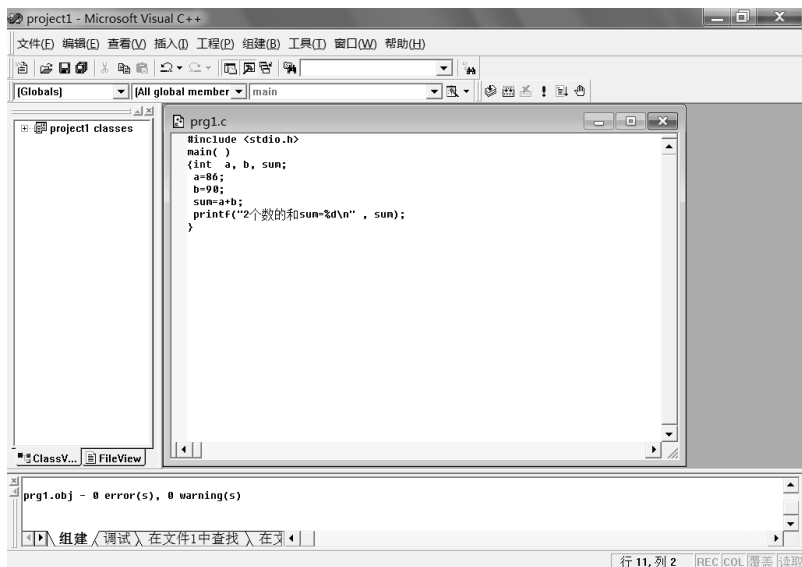



图 1-5 编译完成后的窗口

(2) 链接产生.exe 可执行文件并运行它。编译时没有错误才可进行链接, 单击工具栏上的链接运行按钮, 若链接无错误则显示: 0 error(s), 0 warning(s), 并自动运行程序, 显示如图 1-6 所示的运行程序窗口, 否则显示错误提示信息。同样, 修改后要重新编译、链接、运行。

注意: 要建立另一个新工程, 必须关闭当前区, 单击“文件”菜单→“关闭工作区”, 然后再建立新工程文件。

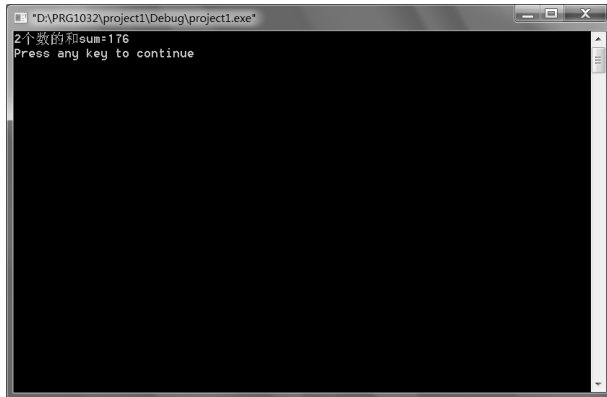


图 1-6 运行程序窗口

1.2 Turbo C++ 3.0 集成开发环境

Turbo C++ 3.0 是美国 Borland 公司在 1992 年推出的强大的 C 语言与 C++ 语言程序设计的集成开发环境。与早期版本不同的是, 它完全支持鼠标选择、拖放和右键操作, 使用更加方便。Turbo C++ 3.0 可在 DOS 和 Windows 两种环境下启动。本书只介绍后者。

1.2.1 启动 Turbo C++ 3.0

启动 Turbo C++ 3.0:

方法一: 如果 Turbo C++ 3.0 已安装到 C 盘的 C:\TC 文件夹, 那么, 打开资源管理器, 双击 C:\TC\BIN 文件夹中的 TC.EXE 文件即可启动, 如图 1-7 所示。

方法二: 如果 Windows 桌面上已建立 TC 快捷方式, 双击该快捷图标即可启动。

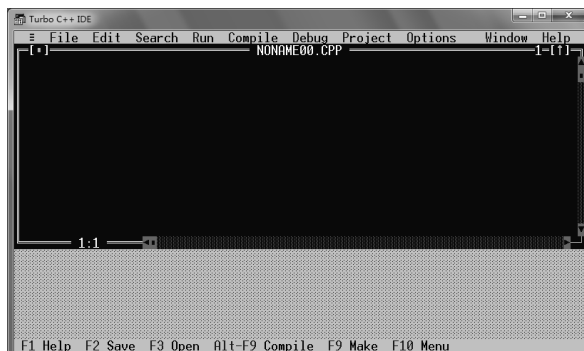


图 1-7 Turbo C++ 3.0 主窗口

开发环境的设置

首先在 D 盘上建立一个文件夹，以便于存放创建、编译、链接将要产生的一系列文件。例如，文件夹 D:\TC1032（TC 后跟学号的末 4 位）。

启动 Turbo C++ 3.0 后，单击“Options”菜单→选择“Directories”命令，按图 1-8 所示进行设置。C:\TC\INCLUDE 是系统包含文件所在的文件夹，C:\TC\LIB 是系统库文件所在文件夹。用户编译、链接产生的文件存放在 D:\TC1032 文件夹中。

注意：当 Turbo C++ 3.0 使用完毕后，要确保下次进入时上述设置仍有效，必须单击“File”菜单→选择“Quit”命令（或按 Alt+X 键）退出 Turbo C++ 3.0。

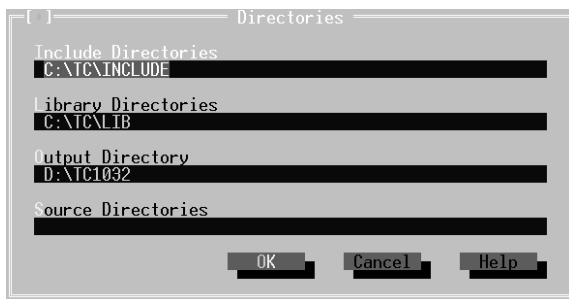


图 1-8 Directories 设置对话框

1.2.2 C 程序的创建、编译、链接、运行

1. 创建新的源程序文件

单击“File”菜单→选择“New”命令→选择“Save”命令→输入文件夹及文件名，例如，C:\TC1032\PRG1.C→输入源程序，如图 1-9 所示。

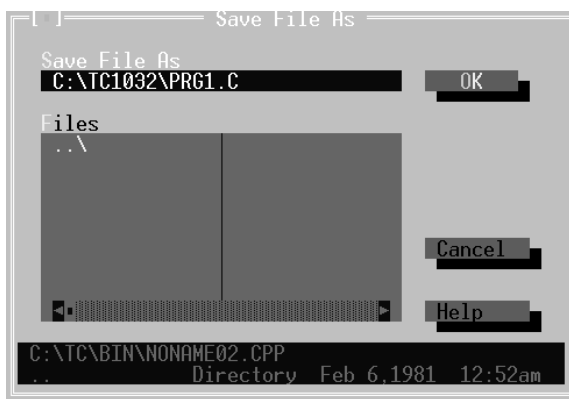


图 1-9 保存新建源程序文件对话框

当输入和修改源程序无误时就可以进行编译了。

2. 编译、链接、运行程序

（1）编译：单击“Compile”菜单→选择“Compile”命令（或按 Alt+F9 键）对源程序进行编译，若编译成功，将产生.obj 目标文件，显示如图 1-10 所示。若有错误将会在下方的信息窗口显示错误提示信息，需修改源程序后再次编译直到无错才可进行链接，如图 1-11 所示。

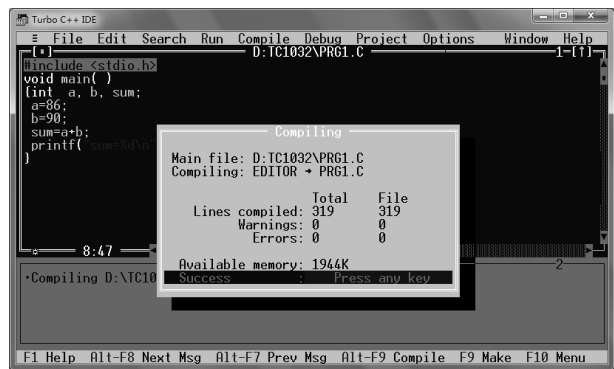


图 1-10 编译成功时的窗口

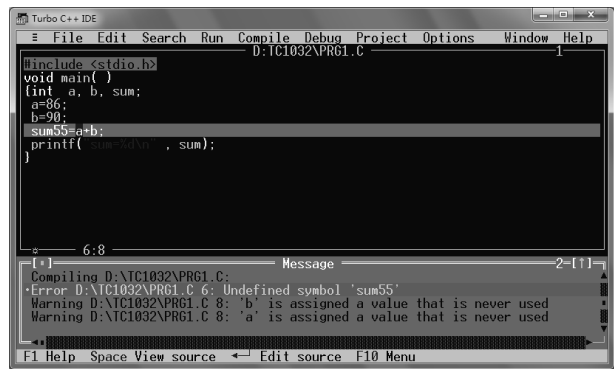


图 1-11 编译出错时的窗口

（2）链接：单击“Compile”菜单→选择“Link”命令，若链接成功会产生一个.exe 可执行文件，若有错误需修改程序后再次编译、链接直到无错才可运行产生的可执行文件。

（3）运行程序：单击“Run”菜单→选择“Run”命令（或按 Ctrl+F9 键）运行程序。切换到运行结果窗口：单击“Window”菜单→选择“User Screen”命令（或按 Alt+F5 键）切换到显示运行结果的窗口。按回车键可切换到程序编辑窗口。

注意：“Run”命令除了可以运行程序之外，还具有编译、链接功能。建立源程序后，可以直接用它编译、链接、运行程序。

Turbo C++ 3.0 常用快捷键的功能如表 1-1 所示。

快捷键	功能
F3	打开一个源程序文件
F2	保存当前编辑的程序
Shift+Del	剪切
Ctrl+Ins	复制
Shift+ Ins	粘贴
Alt+F9	编译
F9	编译并链接
Ctrl+ F9	Run 命令
Alt+F5	切换到运行结果窗口
Alt+X	退出 Turbo C++ 3.0

第2章

C 语言程序设计实验

2.1 熟悉 C 语言程序集成开发环境

2.1.1 同步自测

本书中多项选择题会加以说明，否则都是单选题。

- 编写 C 语言源程序并上机运行的一般过程为（ ）。
A. 编辑、编译、链接和运行 B. 编译、编辑、链接和运行
C. 链接、编译、编辑和运行 D. 链接、编辑、编译和运行
- C 语言程序从（ ）开始执行。
A. 主调函数 B. 自定义函数 C. `main()`函数 D. 第一个函数
- C 语言程序注释可写在（ ）符号内。
A. `/**/` B. `*/` `/*` C. `*` `*` D. `#` `#`
- `main()`函数首部，在 Turbo C++ 3.0 下应写为（ ）。
A. `main()` B. `void main()` C. `main` D. `void main`
- `main()`函数的函数体要用（ ）括起来。
A. 一对小括号 B. 一对中括号 C. 一对大括号 D. 以上都不对
- 每个 C 程序开始要写编译预处理语句，正确的是（ ）。
A. `#include <stdio.h>;` B. `#include stdio.h`
C. `include <stdio.h>` D. `#include <stdio.h>`
- 每个 C 语句必须以（ ）结束。
A. 逗号 B. 分号 C. 句号 D. 以上都不对
- 关于 Visual C++ 6.0 及 Turbo C++ 3.0 开发环境，以下正确的是（ ）。
A. Turbo C++ 3.0 不支持鼠标操作 B. 两种开发环境都不支持鼠标操作
C. Visual C++ 6.0 不支持鼠标操作 D. 两种开发环境都支持鼠标操作
- 以下叙述正确的一项是（ ）。
A. 在 C 语言程序中，`main` 函数必须位于程序的最前面
B. C 语言程序的每一行只能写一条语句
C. 在 C 语言程序编译过程中，可发现注释中的拼写错误
D. C 语言本身没有输入/输出语句

10. 以下叙述正确的一项是 ()。

- A. 一个 C 程序中可以有多个 main() 函数
- B. 调试成功一个 C 程序后, 只能关闭 Visual C++ 6.0, 重新启动才能再调试第二个 C 程序
- C. 调试成功一个 C 程序后, 发“关闭工作区”的命令, 即可再调试第二个 C 程序
- D. C 程序修改后, 不需要再次编译、链接

第 11 题至第 15 题, 用“T”表示正确, “F”表示错误。

11. 判断题: 一个 C 程序不能由多个函数组成。()

12. 判断题: 目标文件.obj 和可执行文件.exe 都可以运行。()




13. 判断题: 先编译生成目标文件.obj, 然后链接生成可执行文件.exe。()

14. 判断题: 有语法错误的某语句编译后, 双击错误提示行, 系统会指明出错语句行。()





15. 判断题: 下列程序中有 7 个错误。()

```
#include <stdio.h>;
main();
{inta, b, sum;
a=86,
b=90;
sum=a+b; /*求和/*
printf("sum=%d\n" , sum);
```

16. 多项选择题: 关于 Visual C++ 6.0 工具栏中的命令按钮, 以下正确的说法是 ()。

- A. 对源程序进行编译的命令按钮是 
- B. 对源程序进行编译的命令按钮是 
- C. 编译、链接、运行程序命令按钮是 
- D. 编译、链接、运行程序命令按钮是 !

17. 多项选择题: 关于 Visual C++ 6.0 工具栏中的命令按钮, 以下正确的说法是 ()。

- A. 剪切命令按钮是 
- B. 复制命令按钮是 
- C. 复制命令按钮是 
- D. 粘贴命令按钮是 

18. 填空题: Visual C++ 6.0 () 菜单中有剪切命令 (填汉字的菜单名)。

19. 填空题: Visual C++ 6.0 中编译命令的快捷键是 (), 编译、链接、运行命令的快捷键是 ()。

20. Visual C++ 6.0 中查找、复制、粘贴命令的快捷键依次是 ()、()、()。

2.1.2 范例程序解析

例 2.1 某人信用卡逾期还款金额为 8630 元, 银行按每天 0.05% 的利率收取利息。计算 25 天之后他的应还款总额。

程序如下:

```
#include <stdio.h>
```

```
main()
{int day;
 float principal, rate , sum;
 principal=8630;           /*逾期还款金额*/
 rate=0.0005;              /*每天利率*/
 day=25;                   /*天数*/
 sum=principal+principal*rate*day; /*计算应还款总额*/
 printf("应还款总额 %.2f\n" , sum); /*显示应还款总额*/
}
```

例 2.2 编程显示 Turbo C++ 3.0 常用快捷键的功能。程序运行结果如图 2-1 所示。

```
#include <stdio.h>
main()
{printf("\t\t\t Turbo C++3.0 常用快捷键\n\n");
 printf("\t\t%-15s%-30s\n" , "快捷键", "功能");
 printf("\t\t%-15s%-30s\n" , "F3", "打开一个源程序文件");
 printf("\t\t%-15s%-30s\n" , "F2", "保存当前编辑的程序");
 printf("\t\t%-15s%-30s\n" , "Shift+Del", "剪切");
 printf("\t\t%-15s%-30s\n" , "Ctrl+Ins", "复制");
 printf("\t\t%-15s%-30s\n" , "Shift+Ins", "粘贴");
 printf("\t\t%-15s%-30s\n" , "Alt+F9", "编译");
 printf("\t\t%-15s%-30s\n" , "F9", "编译并链接");
 printf("\t\t%-15s%-30s\n" , "Ctrl+ F9", "RUN 命令");
 printf("\t\t%-15s%-30s\n" , "Alt+F5", "切换到运行结果窗口");
 printf("\t\t%-15s%-30s\n" , "Alt+X", "退出 Turbo C++3.0");
}
```



图 2-1 例 2.2 程序运行结果

2.1.3 实验项目

1. 实验目的和要求

- (1) 熟悉 Visual C++ 6.0 集成开发环境。
- (2) 在调试程序中观察系统提示，改正编译错误或链接错误。
- (3) 了解 C 语言源程序的结构及书写格式，了解表达 C 语言程序的各个组成成分的术语。

2. 实验内容和步骤

- (1) 首先在 D 盘上建立一个文件夹，以便于存放将要产生的一系列文件。例如文件夹

D:\PRG1032 (PRG 后跟学号的末 4 位)。在该文件夹中新建一个存放实验报告的 Word 文档, 将每次实验的运行结果窗口存放在该文档中。例如, RUN1032.DOC (或 RUN1032.DOCX)。实验报告的格式要求见附录 B。

(2) 上机实验 1

启动 Visual C++ 6.0, 按第 1 章介绍的步骤建立一个 C 语言源程序文件。程序 1 如下:

```
#include <stdio.h>
main()
{int a, b, sum;
 a=86;
 b=90;
 sum=a+b;
 printf("sum=%d\n" , sum);
}
```

注意: 若在 Turbo C++ 3.0 下, 那么 main() 要修改为 void main(), 本书中程序均在中文企业版 Visual C++ 6.0 下调试通过, 以后不再赘述。

(3) 编译、链接、运行程序 1

观察运行结果。按下 Alt+PrintScreen 键截取运行结果窗口图片, 将图片“粘贴”到上述 Word 文档中。

注意: 每次实验都将运行结果窗口图片存入相应实验报告文件中, 后续章节中不再赘述。

(4) 修改程序

将#include <stdio.h>删除;

将 sum=a+b; 语句修改为 sum=55=a+b;, 再编译观察并记录系统提示, 重新改回正确的程序, 编译、链接、运行程序。

(5) 将 printf("sum=%d\n", sum); 语句修改为 printf("两数之和为 %d\n", sum);, 观察运行结果与上次有什么不同。截取运行结果窗口图片存入 Word 文档。

(6) 上机实验 2

关闭工作区, 建立第 2 个 C 语言源程序文件。程序 2 如下:

```
#include <stdio.h>
main()
{printf("\t\t\t Visual C++6.0 常用快捷键\n\n");
 printf("\t\t%-15s%-30s\n" , "快捷键", "功能");
 printf("\t\t%-15s%-30s\n" , "Ctrl+N" , "新建工程文件");
 printf("\t\t%-15s%-30s\n" , "Ctrl+O" , "打开文件");
 printf("\t\t%-15s%-30s\n" , "Ctrl+F7" , "编译");
 printf("\t\t%-15s%-30s\n" , "Ctrl+F5" , "编译链接运行");
 printf("\t\t%-15s%-30s\n" , "Alt+F8" , "选中代码书写格式对齐");
}
```

(7) 编译、链接、运行程序 2

观察运行结果, 截取运行结果窗口图片存入 Word 文档。

(8) 修改程序 2

在第 5 个 printf 语句之前插入 5 个新语句, 分别显示下列快捷键:

Ctrl+X 剪切

Ctrl+C 复制

Ctrl+V 粘贴

Del 删除

Ctrl+F 查找

编译、链接、运行修改后的程序，截取运行结果窗口图片存入 Word 文档。

(9) 裁剪运行结果图片

将 Word 文档中的运行结果窗口图片用 Word 的图片“裁剪”命令按钮，裁掉多余部分，只保留显示运行结果的那部分。

注意：每次实验都将运行结果窗口图片存入相应实验报告文件中，后续章节中不再赘述。

3. 实验思考和总结

(1) 调试通过后产生的可执行文件在哪个文件夹中？文件名与工程名相同，还是与源程序文件名相同？

(2) 本次实验你积累了哪些经验？有何体会？

2.2 数据类型、运算符、表达式

2.2.1 同步自测

1. 在 C 语言中，要求参加运算的数必须是整数的运算符是 ()。
A. / B. * C. % D. =
2. 下面四个选项中均是合法的整型常量的是 ()。
A. 340 B. 02 C. -0X2A D. 0X15
025 0X 985,768 0X2
-11 03f 4d2 -760X
3. 变量 i, j 为整型变量，表达式(i=2, j=5, j++, i+j)的值是 ()。
A. 7 B. 4 C. 8 D. 5
4. 若定义 int k=5; 表达式 k+=k-=k*k 的值是 ()。
A. 38 B. -40 C. -42 D. 40
5. 若 int a=5;，使 b 不为 2 的表达式是 ()。
A. b=a/2 B. b=6-(-a)
C. b=a%2 D. b=a>3?2:1
6. 若 int k=7, x=12; 则能使值为 3 的表达式是 ()。
A. x%=(k%=5) B. x%=(k-k%5)
C. x%=k-k%5 D. (x%=k)-(k%=5)
7. 表示代数式 $\frac{2ab}{cd}$ 的 C 语言错误的表达式是 ()。
A. a/c/d*b*2 B. 2*a*b/c*d C. 2*a*b/c/d D. a*b/c/d*2

8. 下列标识符中不合法是 ()。
- A. apple_2 B. cat1 C. 2apple D. _cat1
9. 若变量定义并赋值, 以下合法的赋值语句是 ()。
- A. x=n%2.5; B. x=y=25; C. x=25=24+1; D. n+x=25;
10. 以下所列的 C 语言常量中, 错误的是 ()。
- A. 0xFF B. 1.2e0.5 C. 2L D. '\72'
11. 多项选择题: 以下合法的字符常量是 ()。
- A. '\t' B. "A" C. 'a' D. '\x32'
12. 多项选择题: 下列语句中, 符合语法的赋值语句是 ()。
- A. a=7+b+c=a+7; B. a=7+b++=a+7;
C. a=(7+b,b++,a+7); D. a=7+b;c=a+7;
13. 多项选择题: 以下合法的赋值语句是 ()。
- A. x=y=100; B. d--; C. x+y; D. c=int(a+b);
14. 填空题: 若有语句 double x = 17; int y; 当执行 y=(int)(x/5)%2; 之后 y 的值是 ()。
15. 填空题: 已知字母 A 的 ASCII 码值为 65。下列程序运行后的输出结果是 ()。

```
#include <stdio.h>
main()
{char a,b;
 a='A'+ '5' - '3';
 b=a+ '6' - '2';
 printf("%d%c\n", a,b);
}
```

16. 填空题: 下列程序运行后的输出结果是 ()。

```
#include <stdio.h>
main()
{char a='1' ,b='2';
 printf("%c#", b++);
 printf("%d\n", b-a);
}
```

17. 判断题: C 语言不允许混合类型数据间进行运算。()
18. 判断题: 任何一个实型变量的值肯定是精确的。()
19. 判断题: 自增运算符 (++) 或自减运算符 (--) 只能用于变量, 不能用于常量。()
20. 判断题: 对几个变量在定义时赋初值可以写成: int a=b=c=3;。()

2.2.2 范例程序解析

例 2.3 求 2 个 10 位数相加之和: 1234567890+2000000000。

程序如下:

```
#include <stdio.h>
main()
{long a,b,sum;
 a=1234567890;
```



```
b=2000000000;
sum=a+b;
printf("sum=%ld\n", sum);
}
```

运行发现结果是一个负数，说明发生了数据溢出。

将其修改为如下程序，将变量定义为 **double** 型，即可得到正确的运行结果。

```
#include <stdio.h>
main()
{double a,b, sum;
a=1234567890;
b=2000000000;
sum=a+b;
printf("sum=%.0f\n", sum); /*%.0f 只输出整数*/
}
```

运行结果：

```
sum=3234567890
```

例 2.4 字符型变量值的输出。

程序如下：

```
#include <stdio.h>
main ()
{char c1 ,c2;
c1='D';
c2=c1+1;
printf("%c %c\n" , c1,c2); /*显示大写字母 D E*/
printf("%d %d\n" , c1,c2); /*显示大写字母 D E 的 ASCII 码*/
}
```

运行结果：

```
D E
68 69
```

思考：c1='D'；可以修改为 c1=68；吗？

例 2.5 自增、自减运算举例。

程序如下：

```
#include <stdio.h>
main ()
{int i,j,k1,k2;
i=11;
j=21;
k1=i++; /*首先将 i 原值 11 赋值给 k1,然后 i 自增为 12*/
k2=j--; /*首先将 j 原值 21 赋值给 k2,然后 j 自减为 20*/
printf("%d %d %d %d\n" , i,j,k1,k2);
i=11;
j=21;
k1=++i; /*首先 i 自增为 12,并将 12 赋值给 k1*/
```

```
k2=--j;           /*首先 j 自减为 20, 并将 20 赋值给 k2*/
printf("%d %d %d %d\n" , i,j,k1,k2);
}
```

运行结果:

```
12 20 11 21
12 20 12 20
```

例 2.6 复合运算符、逗号运算符应用举例。

程序如下:

```
#include <stdio.h>
main ()
{int m=15, n=9, s=0;
 m*=3+2;
 printf("%d\n" , m);
 n+=n-=n*=n ;           /*先计算 n=n*n; 然后计算 n=n-n; 最后计算 n=n+n;*/
 printf ("%d\n", n);
 printf ("%d\n" , (s=6*5,m+n+s)); /*显示逗号表达式的值*/
 printf ("%d\n" , s);
}
```

运行结果:

```
75
0
105
30
```

例 2.7 表达式的数据类型举例。

程序如下:

```
#include <stdio.h>
main ()
{int a1=7,b1=3;
 float x=2.5,y=4.7,a2=7,b2=3;
 printf("%d\n",a1/b1); /*a1、b1 为 int 型, 则 a1/b1 为 int 型的值*/
 printf("%.15f\n",a2/b2); /*a2/b2 为 double 型的值,%.15f 显示 15 位小数*/
 printf("%f\n",x+a1%3*(int)(x+y)%2/4);
 printf("%f\n", (float)(a1+b1)/2+(int)x%(int)y);
}
```

运行结果:

```
2
2.3333333333333334
2.500000
7.000000
```

2.2.3 实验项目

1. 实验目的和要求

- (1) 掌握 C 语言的基本数据类型, 理解数据溢出的概念。
- (2) 掌握 C 语言的算术运算符、赋值运算符、自增自减运算符、逗号运算符的含义及运算规则。
- (3) 掌握表达式的求值规则。

2. 实验内容和步骤

(1) 输入并运行下列程序，它可以显示 1~15 的阶乘。

```
#include <stdio.h>
main()
{int i;
 double t=1;
 for (i=1; i<=15; i++)
 {t=t*i;
 printf("%d!=%.0f\n", i, t); }
}
```

记下 13、14、15 的阶乘结果，然后进行如下修改：

double t=1;	修改为	long t=1;
printf("%d!=%.0f\n",i,t);	修改为	printf("%d!=%ld\n",i,t);

运行修改后的程序，观察发现 13、14、15 的阶乘结果与上次完全不同，为什么？

(2) 输入并运行下列程序，了解字符型数据和整型数据之间的关系。

```
#include <stdio.h>
main ()
{char c1 ,c2;
 c1=65;
 c2=66;
 printf("%c %c\n" , c1,c2);
 printf("%d %d\n" , c1,c2);
}
```

输入并运行下列程序，用所学的知识解释运行结果。

```
#include <stdio.h>
main()
{int i, j, k,a1=17,b1=2;
 float a2=17,b2=2;
 i=5;
 j=++i;
 k=--i;
 printf("i=%d j=%d k=%d\n",i, j, k);
 k=i++;
 j=(--i)+k;
 printf("i=%d j=%d k=%d\n",i, j, k);
 printf("%d\n" , (i,j,k));
 printf("%d\n" , a1/b1);
 printf("%d\n" , a1%b1);
 printf("%f\n" , a2/b2);
 printf("%f\n" , a1/b2);
}
```

3. 实验思考和总结

- (1) 为什么修改了存放阶乘的变量 t 的数据类型, 求出的 13、14、15 的阶乘结果就变了?
- (2) 在第 3 个 C 程序中为什么表达式 $a1/b1$ 和 $a2/b2$ 显示的结果不同?
- (3) 在第 3 个 C 程序中, 若赋值 $a1=a2/b2$; 则 $a1$ 的值是 8 还是 8.5?
- (4) 在第 3 个 C 程序中, 若赋值 $a2=a1/b1$; 则 $a2$ 的值是 8 还是 8.5?

2.3 顺序结构程序设计

2.3.1 同步自测

1. 有如下程序

```
#include <stdio.h>
main()
{int a; char c=10;
 float f=100.0; double x;
 a=f/=c*=(x=6.5);
 printf("%d %d %3.1f%3.1f\n",a,c,f,x);
}
```

程序运行后输出结果是 ()。

- A. 1 65 1 6.5 B. 1 65 1.5 6.5 C. 1 65 1.0 6.5 D. 2 65 1.5 6.5

2. 设变量 x 为 float 型且已赋值, 则以下语句中能将 x 中的数值保留到小数点后两位, 并将第三位四舍五入的是 ()。

- A. $x=x*100+0.5/100.0$ B. $x=(x*100+0.5)/100.0$
C. $x=(int)(x*100+0.5)/100.0$ D. $x=(x/100+0.5)*100.0$

3. 有下列程序

```
#include <stdio.h>
main()
{int a=0,b=0;
 a=10;
 b=20;
 printf("a+b=%d\n",a+b);
}
```

程序运行后的输出结果是 ()。

- A. $a+b=10$ B. $a+b=30$ C. 30 D. 出错

4. 若有定义 $\text{int } a=8, b=5, c;$, 执行语句 $c=a/b+0.4$; 后 c 的值是 ()。

- A. 1.4 B. 1 C. 2.0 D. 2

5. 若有以下定义和语句:

```
int u=010, v=0x10, w=10;
printf("%d,%d,%d\n",u,v,w);
```

则输出结果是 ()。

A. 8,16,10

B. 10,10,10

C. 8,8,10

D. 8,10,10

6. 下列程序的输出结果是 ()。

```
#include <stdio.h>
main( )
{ int x=17;
printf("%d,%o,%x \n",x, x, x); }
```

A. 17,021,0x11

B. 17,17,17

C. 17,0x11,021

D. 17,21,11

7. 有下列程序

```
#include <stdio.h>
main ()
{char a1='M', a2='m';
printf("%c\n", (a1,a2)); }
```

以下叙述中正确的是 ()。

A. 程序输出大写字母 M

B. 程序输出小写字母 m

C. 格式说明符不足, 编译出错

D. 程序运行时产生出错信息

8. 若变量为 float 类型, 用语句 scanf("%f%f%f",&a,&b,&c), 依次给 a,b,c 赋值 77,88,99。

以下不正确的输入方法是 ()。

A. 77✓

B. 77,88,99✓

88✓

99✓

C. 77✓

D. 77 88✓

88 99✓

99✓

9. 有输入语句: scanf("a=%d,b=%d,c=%d",&a,&b,&c);。为使变量 a 的值为 1, b 的值为 3, c 的值为 2, 则正确的数据输入方式是 ()。

A. 132✓

B. 1,3,2✓

C. a=1 b=3 c=2✓

D. a=1,b=3,c=2✓

10. 若运行时输入: 12345678✓, 则下列程序运行结果为 ()。

```
#include <stdio.h>
main()
{ int a,b;
scanf("%2d%2d%3d",&a,&b);
printf("%d\n",a+b); }
```

A. 46

B. 579

C. 5690

D. 出错

11. 多项选择题: 若 int x,y; double z;则以下不合法的 scanf 函数调用是 ()。

A. scanf("%d%lx,%le",&x,&y,&z);

B. scanf("%2d*%d%f",&x,&y,&z);

C. scanf("%d %*d %d",&x,&y);

D. scanf("%x%o%6.2f", &x,&y,&z);

12. 判断题: printf 和 scanf 语句中 long 类型对应的格式说明符都是%ld。()
13. 判断题: printf 和 scanf 语句中 double 类型对应的格式说明符都是%f。()
14. 判断题: 若输入语句 scanf("input k? %f", &k);则屏幕上会自动显示“input k?”。()
15. 判断题: putchar(getchar());是正确的语句。()
16. 判断题: scanf("%f\n", &k);其中\n 起换行作用。()
17. 判断题: scanf("%f, %f", &k1, &k2);输入的两个数据之间要输入空格。()
18. 判断题: printf 函数不能显示八进制数。()
19. 判断题: printf 函数中用%f 只能显示 float 类型的数据。()
20. 判断题: 从键盘上给字符型变量赋值只能使用 scanf 语句。()

2.3.2 范例程序解析

例 2.8 常用数据类型输出方法举例 1。

程序如下:

```
#include <stdio.h>
main()
{int x1,y1;
 unsigned int x2,y2;
 long x3,y3;
 char x4,y4;
 float x5,y5;
 double x6,y6;
 x1=11/2;      y1=17/3;
 x2=56000;     y2=x2+30000;
 x3=654321;    y3=-842100;
 x4='a';       y4=98;
 x5=11.0/2;    y5=17.0/3;
 x6=11.0/2;    y6=17.0/3;
 printf("x1=%d\ty1=%d\n", x1, y1);
 printf("x1=%f\ty1=%f\n", x1, y1); /*使用了错误的格式说明符%f*/
 printf("x2=%u\ty2=%u\n", x2, y2);
 printf("x3=%ld\ty3=%ld\n", x3, y3);
 printf("x4=%c\ty4=%c\n", x4, y4);
 printf("x4=%d\ty4=%d\n", x4, y4);
 printf("x5=%f\ty5=%f\n", x5, y5);
 printf("x5=%.13f\ty5=%.13f\n", x5, y5);
 printf("x6=%.13f\ty6=%.13f\n", x6, y6);
 printf("x6=%f\ty6=%f\n", x6, y6);
}
```

运行结果:

```
x1=5      y1=5
x1=0.000000    y1=0.000000
```

```

x2=56000          y2=86000
x3=654321         y3=-842100
x4=a              y4=b
x4=97             y4=98
x5=5.500000       y5=5.666667
x5=5.500000000000 y5=5.6666665077209
x6=5.500000000000 y6=5.6666666666667
x6=5.500000       y6=5.666667

```

例 2.9 常用数据类型输出方法举例 2。

程序如下：

```

#include <stdio.h>
main()
{unsigned int  x1=65535;
  int  x2=-3;
  float y1=123.456,y2=13.45;
  printf("x1=%d,%o,%x,%u\n" ,x1,x1,x1,x1);
  printf("x2=%d,%o,%x,%u\n" ,x2,x2,x2,x2);
  printf("y1=%10f,%10.3f,%.3f,%-10.3f\n" ,y1,y1,y1,y1);
  printf("y2=%f,%e,%g\n",y2,y2,y2);
  printf("%s\n" , (1,2.0,'?',"abc"));
}

```

运行结果：

```

X1=65535,177777,ffff,65535
X2=-3,37777777775,ffffffffff,4294967293
Y1=123.456001, 123.456,123.456,123.456
Y2=13.450000,1.345000e+001,13.45
abc

```

输出 y2 的值时 %e 表示以指数形式输出。

%g 表示系统自动选 %f 格式或 %e 格式中长度较短的格式，不输出无意义的 0。

例 2.10 编程从键盘输入圆柱体的半径和高，求它的表面积和体积。

程序如下：

```

#define PI 3.14159
#include <stdio.h>
main()
{float r,h,area,volume;
  printf("请输入圆柱体的半径和高");
  scanf("%f%f" ,&r,&h);      /*输入的两个数之间须用空格隔开*/
  area=2*PI*r*h+2*PI*r*r;    /*计算表面积*/
  volume=PI*r*r*h;           /*计算体积*/
  printf("圆柱体的表面积%.2f    体积%.2f\n" ,area,volume);
}

```

输入方法正确时的运行结果：

```

请输入圆柱体的半径和高 3.5 7.2
圆柱体的表面积 235.31    体积 277.09

```

输入方法错误时的运行结果：（输入的两个数之间用了逗号）

请输入圆柱体的半径和高 3.5, 7.2
圆柱体的表面积-2361279488.00 体积-4132239060.35

例 2.11 某人网购了 3 种商品，单价、数量、运费如下，编程从键盘输入各商品的单价、数量及运费，计算每种商品的消费额和消费总额。要求保留 2 位小数。

商品名	单价	数量	运费
茶树菇	24.80	6 袋	10
蜂蜜	88.0	2 瓶	0
笋尖	19.90	7 袋	10

程序如下：

```
#include <stdio.h>
main()
{float price,number,freight,money1,money2,money3,money;
printf("请输入茶树菇的单价 数量 运费" );
scanf("%f%f%f" ,&price,&number,&freight);
money1=price*number+freight;          /*计算茶树菇的消费额*/
printf("请输入蜂蜜的单价 数量 运费" );
scanf("%f%f%f" ,&price,&number,&freight);
money2=price*number+freight;          /*计算蜂蜜的消费额*/
printf("请输入笋尖的单价 数量 运费" );
scanf("%f%f%f" ,&price,&number,&freight);
money3=price*number+freight;          /*计算笋尖的消费额*/
money=money1+money2+money3;          /*计算总消费额*/
printf("茶树菇的消费额%.2f 元\n" , money1);
printf("蜂蜜的消费额%.2f 元\n" , money2);
printf("笋尖的消费额%.2f 元\n" , money3);
printf("总消费额%.2f 元\n" , money);
}
```

运行结果：

请输入茶树菇的单价 数量 运费 24.8 6 10
请输入蜂蜜的单价 数量 运费 88 2 0
请输入笋尖的单价 数量 运费 19.9 7 10
茶树菇的消费额 158.80
蜂蜜的消费额 176.00
笋尖的消费额 149.30
总消费额 484.10

2.3.3 实验项目

1. 实验目的和要求

- （1）学会编写简单顺序结构的程序。
- （2）掌握各种类型数据的输入输出方法。
- （3）掌握编写和调试程序的方法。

2. 实验内容和步骤

(1) 输入下列程序，掌握各种类型变量正确的输出方法。

```
#include <stdio.h>
main()
{int a1,b1;
 unsigned int a2,b2;
 long a3,b3;
 char a4,b4;
 float a5,b5;
 double a6,b6;
 a1=15/2;      b1=7/3;
 a2=45768;     b2=a2+20000;
 a3=54321;     b3=-67000;
 a4='A';       b4=66;
 a5=15.0/2;    b5=7.0/3;
 a6=15.0/2;    b6=7.0/3;
 printf("a1=%d\tb1=%d\n",a1,b1);
 printf("a1=%f\tb1=%f\n",a1,b1);
 printf("a2=%u\tb2=%u\n",a2,b2);
 printf("a3=%ld\tb3=%ld\n",a3,b3);
 printf("a4=%c\tb4=%c\n",a4,b4);
 printf("a4=%d\tb4=%d\n",a4,b4);
 printf("a5=%f\tb5=%f\n",a5,b5);
 printf("a5=%.13f\tb5=%.13f\n",a5,b5);
 printf("a6=%.13f\tb6=%.13f\n",a6,b6);
 printf("a6=%f\tb6=%f\n",a6,b6);
}
```

思考：为什么运行结果第2行显示 a1,b1 的值是错误的？

计算 7.0/3 用什么类型的变量存放更精确？

(2) 两个同心圆构成一个空心圆环，编程从键盘输入大圆的半径 12.5cm 和小圆的半径 8.2cm，求圆环的面积。要求定义符号常量表示圆周率，运行结果保留 2 位小数。可参考本节范例程序例 2.10。

思考：从键盘输入两个数的方法与 scanf 语句的写法有关吗？

(3) 某顾客从超市采购袋装食品如下，编程从键盘输入各商品的单价和数量，计算每种商品的消费额和消费总额。要求保留 2 位小数。可参考本节范例程序例 2.11。

商品名	单价	数量（袋）
白糖	13.52 元	2
蛋糕	16.8 元	5
红枣	22.3 元	3

思考：单价、消费额、数量（袋数）应分别定义什么类型的变量？

3. 实验思考和总结

(1) 总结编程调试的经验教训。

(2) 回答上述各思考题。

2.4 选择结构程序设计

2.4.1 同步自测

1. 能正确表示“当 x 的取值在[1,10]和[200,210]范围内时为真, 否则为假”的表达式是 ()。

- A. $(x>=1) \&\& (x<=10) \&\& (x>=200) \&\& (x<=210)$
- B. $(x>=1) || (x<=10) || (x>=200) || (x<=210)$
- C. $(x>=1) \&\& (x<=10) || (x>=200) \&\& (x<=210)$
- D. $x>=1 || (x<=10) \&\& (x>=200) || (x<=210)$

2. 若 x, y, z 是整型变量, 且 $x=1 \quad y=2 \quad z=3$ 。

- (1) $x\&\&y\&\&z$ (2) $x>=y$ (3) $x|y+z\&\&x$ (4) $!((x<y)\&\&!z|1)$

以上 4 个表达式的值依次是 ()。

- A. 1, 0, 2, 1 B. 1, 0, 1, 0 C. 0, 1, 1, 0 D. 1, 0, 2, 0

3. 以下程序输出结果是 ()。

```
#include <stdio.h>
main()
{int x=200, y=-10, z=300;
  if (x<y)
  if (y<0) z=-z;
  else z+=10;
  printf("%d\n", z); }
```

- A. 0 B. -300 C. 200 D. 300

4. 以下程序输出结果是 ()。

```
#include <stdio.h>
main()
{int a=8, b=9, t;
  t=a;
  if (a<b) t=b;
  t*=t;
  printf("%d \n" , t); }
```

- A. 8 B. 9 C. 81 D. 64

5. 在下面的条件语句中 (其中 S1 和 S2 表示 C 语言语句), 只有一个在功能上与其他三个语句不等价的是 ()。

- A. if (a) S1; else S2; B. if (a==0) S2; else S1;
- C. if (a!=0) S1; else S2; D. if (a==0) S1; else S2;

6. 在嵌套使用 if 语句时, C 语言规定 else 总是 ()。

- A. 和之前与其具有相同缩进位置的 if 配对
- B. 和之前与其最近的 if 配对
- C. 和之前与其最远的 if 配对
- D. 和之前的第一个 if 配对

7. 以下程序运行后的输出结果是 ()。

```
#include <stdio.h>
main()
{int a=2,b=-1,c=2;
  if (a<b)
  if (b<0) c=0;
  else c++;
  printf("%d\n",c);
}
```

- A. 0 B. 1 C. 2 D. 3

8. 以下程序运行后的输出结果是 ()。

```
#include <stdio.h>
main()
{int x1=1,x2=0,x3=0;
  if(x1=x2+x3) printf("*****");
  else printf("####");
}
```

- A. #### B. **** C. 有语法错误 D. 无输出结果

9. 下列叙述中正确的是 ()。

- A. break 语句只能用于 switch 语句
B. 在 switch 语句中必须使用 default
C. break 语句必须与 switch 语句中的 case 配对使用
D. 在 switch 语句中不一定使用 break 语句

10. 以下程序运行后的输出结果是 ()。

```
#include<stdio.h>
main()
{int x=1,y=0,a=0,b=0;
  switch(x)
  {case 1:
    switch(y)
    {case 0: a++; break;
     case 1: b++; break;
    }
    case 2: a++;b++; break;
  }
  printf("a=%d,b=%d\n",a,b);
}
```

- A. a=2,b=1 B. a=1,b=1 C. a=1,b=0 D. a=2,b=2

11. 填空题: 运行下面程序时, 若从键盘输入 7,9, 则程序的输出结果是 ()。

```
#include <stdio.h>
```

```
main()
{int x,y;
 scanf("%d,%d",&x,&y);
 if (x==y)
 printf("x==y");
 else if (x>y)
 printf("x>y");
 else
 printf("x<y");
}
```

12. 填空题：运行以下程序时，从键盘输入字符‘B’和回车键，输出结果是（ ）。

```
#include <stdio.h>
main()
{int k;
 char c;
 c=getchar();
 if (c>='0' && c<='9')
 k=c-'0';
 else if (c>='a' && c<='f')
 k=c-'a'+10;
 else k=c-'A'+10;
 printf("%d\n",k);
}
```

13. 填空题：运行以下程序时，从键盘输入 9，输出结果是（ ）。

```
#include <stdio.h>
main()
{int a ;
 scanf("%d",&a);
 if (a++>9) printf("%d\n",a);
 else printf("%d\n",a--) ;
}
```

14. 填空题：运行以下程序时，从键盘输入 95，输出结果是（ ）。

```
#include <stdio.h>
main()
{int d,i,j,k,p;
 scanf("%d",&d);
 if (d>99)
 p=3;
 else if (d>9)
 p=2;
 else
 p=1;
 i=d/100;
 j=(d-i*100)/10;
```

```
k=(d-i*100-j*10);
switch (p)
{case 3: printf("%d%d%d\n",k,j,i);
        break;
 case 2: printf("%d%d\n",k,j);
        break;
 case 1: printf("%d\n",k); }
}
```

15. 填空题：运行以下程序时，从键盘输入 3,8,5，输出结果是（ ）。

```
#include <stdio.h>
main()
{int a,b,c;
 scanf("%d,%d,%d",&a,&b,&c);
 if (a<b)
 if (b<c)printf("%d\n",c);
 else printf("%d\n",b);
 else if (a<c)printf("%d\n",c);
 else printf("%d\n",a);
}
```

16. 判断题：运行以下程序时，从键盘输入 ‘a’，输出结果是 apple。（ ）

```
#include <stdio.h>
main()
{char c;
 c=getchar();
 switch(c)
 {case 'a':printf("apple\n");
 case 'b':printf("book\n");
 default :printf("computer\n");
 }
}
```

17. 判断题：以下两个程序段的作用相同。（ ）

第1段程序：

```
if (表达式1) 语句1;
if (表达式2) 语句2;
语句3;
```

第2段程序：

```
if (表达式1) 语句1;
else if (表达式2) 语句2;
else 语句3;
```

18. 判断题：运行以下程序时，从键盘输入 5,7,3，输出结果是 3,5,7。（ ）

```
#include <stdio.h>
main()
```

```

{int  a,b,c,t;
scanf("%d,%d,%d", &a,&b,&c);
if(a<b)
    {t=a;a=b;b=t;}
if(a<c)
    {t=a;a=c;c=t;}
if(b<c)
    {t=b;b=c;c=t;}
printf("%d,%d,%d\n",a,b,c);
}

```

19. 判断题：以下程序段能正确计算此函数的值。()

```

y= -1    x<0
y=0      x=0
y=1      x>0

if (x>=0)
    if (x>0) y=1;
else y=0;
else y=-1;

```

20. 判断题：若 `c` 为字符型变量，已知空格 ASCII 码为 32，判断 `c` 是否为空格可以使用 `if (c=='32')` 语句。()

2.4.2 范例程序解析

例 2.12 编程从键盘输入三条边长，判断其是否能构成三角形。若能构成三角形，再判断是哪种类型的三角形（等边三角形、等腰三角形、直角三角形、普通三角形）。并求三角形面积。要求保留 2 位小数。

程序如下：

```

#include <stdio.h>
#include <math.h>
main()
{float a, b, c;
float s, area;
printf("请输入三条边长");
scanf("%f%f%f", &a, &b, &c);          /*输入三条边长*/
if (a+b>c&&b+c>a&&a+c>b)              /*判断两边之和是否大于第三边*/
    {s=(a+b+c)/2;
    area=(float)sqrt(s*(s-a)*(s-b)*(s-c)); /*计算三角形的面积*/
    printf("面积是:%.2f\n", area);      /*输出三角形的面积*/
    if (a==b&&a==c)                    /*判断三条边是否相等*/
        printf("等边三角形\n");
    else if (a==b||a==c||b==c)          /*判断三角形中是否有两边相等*/
        printf("等腰三角形\n");
    else if ((a*a+b*b==c*c)|| (a*a+c*c==b*b)|| (b*b+c*c==a*a))
        /*判断是否两边的平方和等于第三边的平方*/
        printf("直角三角形\n");
    else printf("普通三角形\n");
}

```

```
    }  
    else  
        printf("不能构成三角形\n"); /*两边之和不大于第三边, 不能组成三角形*/  
}
```

第1次运行结果:

```
请输入三条边长 3 6 7  
面积是:8.94  
普通三角形
```

第2次运行结果:

```
请输入三条边长 6 6 8  
面积是:17.89  
等腰三角形
```

第3次运行结果:

```
请输入三条边长 3 4 5  
面积是:6.00  
直角三角形
```

第4次运行结果:

```
请输入三条边长 3.6 3.6 3.6  
面积是:5.61  
等边三角形
```

第5次运行结果:

```
请输入三条边长 3 4 1  
不能构成三角形
```

例 2.13 某省居民阶梯电价分为以下三挡:

第一挡电量: 每户年用电量 ≤ 2520 度, 收费每度 0.55 元;

第二挡电量: 2520 度 $<$ 每户年用电量 ≤ 4800 度, 收费每度 0.60 元;

第三挡电量: 每户年用电量 > 4800 度, 收费每度 0.85 元。

编程从键盘输入一户的全年用电量计算应缴电费, 要求保留 2 位小数。

程序如下:

```
#include <stdio.h>  
main()  
{float number,money;  
    printf("请输入一户的全年用电量");  
    scanf("%f", &number);  
    if (number<=2520) money=number*0.55; /*年用电量 $\leq 2520$ 度*/  
    else if (number<=4800) money=2520*0.55+(number-2520)*0.6;  
    /*2520度 $<$ 年用电量 $\leq 4800$ 度*/  
    else money=2520*0.55+(4800-2520)*0.6+(number-4800)*0.85; /*年用电量 $> 4800$ 度*/  
    printf("用电量%.2f度    电费%.2f元\n", number,money);  
}
```

第 1 次运行结果:

请输入一户的全年用电量 2302
用电量 2302.00 度 电费 1266.10 元

第 2 次运行结果:

请输入一户的全年用电量 4507
用电量 4507.00 度 电费 2578.20 元

第 3 次运行结果:

请输入一户的全年用电量 5328
用电量 5328.00 度 电费 3202.80 元

例 2.14 某市中考体育考试项目之一是掷实心球，考试的实际测试成绩换算成体育考试分数的标准如下：

男	女	分数
9.6 米以上	7.20 米以上	10 分
7.70~9.59 米	6.60~7.19 米	8 分
5.30~7.69 米	6.00~6.59 米	6 分
3.00~5.29 米	4.00~5.99 米	4 分
2.99 米以下	3.99 米以下	2 分

编程从键盘输入一个考生的性别和实际测试成绩，输出其对应的考试分数。性别可用‘m’表示男，‘w’表示女。

程序如下：

```
#include <stdio.h>
main()
{float distance;
char sex;
int score;
printf("请输入性别和投掷实心球实测成绩");
scanf("%c%f", &sex,&distance);
if (sex=='m' || sex=='M')
    if (distance>=9.6) score=10; /*以下 5 行代码求男生的分数 score*/
    else if (distance>=7.7) score=8;
    else if (distance>=5.3) score=6;
    else if (distance>=3.0) score=4;
    else if (distance>0) score=2;
    else {printf("输入的实测成绩有错误，请重新运行程序\n");
return (0);} /*结束程序运行*/
else if (sex=='w' || sex=='W')
    if (distance>=7.2) score=10; /*以下 5 行代码求女生的分数 score*/
    else if (distance>=6.6) score=8;
    else if (distance>=6.0) score=6;
    else if (distance>=4.0) score=4;
    else if (distance>0) score=2;
    else {printf("输入的实测成绩有错误，请重新运行程序\n");
```



```

        return (0);}
    else {printf("输入的性别有错误, 请重新运行程序\n");
        return (0);}
    printf("性别%s    ", (sex=='m' || sex=='M')? "男": "女");
    /*输出条件表达式的值: 男或女*/
    printf("实测成绩%.2f 米    分数%d\n", distance, score); /*输出实测成绩和分数*/
}

```

第1次运行结果:

请输入性别和投掷实心球实测成绩 w 6.48
 性别女 实测成绩 6.48 分数 6

第2次运行结果:

请输入性别和投掷实心球实测成绩 m 9.83
 性别男 实测成绩 9.83 分数 10

第3次运行结果: (输入了负数)

请输入性别和投掷实心球实测成绩 w -7.8
 输入的实测成绩有错误, 请重新运行程序

例 2.15 某公司销售人员的实发工资由底薪 1000 元和销售额提成两部分组成。具体提成比例如下:

销售额	提成比例
销售额 \leq 3000 元	无提成
3000 元 $<$ 销售额 \leq 6000 元	7%
6000 元 $<$ 销售额 \leq 10000 元	10%
销售额 $>$ 10000 元	13%

已知销售额单位为元, 是一个整数。用 switch 语句编程, 从键盘输入一个销售额, 计算该员工的实发工资。要求保留 2 位小数。

程序如下:

```

#include <stdio.h>
main()
{int n,sales;
 float salary=1000;
 printf("请输入员工月销售额: ");
 scanf("%d",&sales);
 if (sales<0) {printf("输入的销售额有错误, 请重新运行程序\n"); return (0);}
 if (sales%1000==0) /*判断销售额/1000 是否为整数*/
     n=sales/1000; /*销售额/1000 是整数时, 用此公式求 n 的值*/
 else
     n=sales/1000+1; /*销售额/1000 不是整数时, 用此公式求 n 的值*/
 switch(n)
 {case 0:
 case 1:
 case 2:
 case 3: salary = salary +0; break; /*销售额 $\leq$ 3000 元无提成时的实发工资*/

```

```
case 4:
case 5:
case 6:
    salary=salary+sales*0.07;    /*3000 元<销售额≤6000 元时的实发工资*/
    break;
case 7:
case 8:
case 9:
case 10:
    salary+=sales*0.10;          /*6000 元<销售额≤10000 元时的实发工资*/
    break;
default: salary=salary+sales*0.13;    /*销售额>10000 元时的实发工资*/
    break;
}
printf("该员工实发工资为: %.2f 元\n", salary);    /*输出员工的实发工资*/
}
```

第 1 次运行结果:

请输入员工月销售额: 3560
该员工实发工资为: 1249.20 元

第 2 次运行结果:

请输入员工月销售额: 9673
该员工实发工资为: 1967.30 元

第 3 次运行结果: (输入负数)

请输入员工月销售额: -5000
输入的销售额有错误, 请重新运行程序

例 2.16 判别键盘输入字符的类别。根据输入字符的 ASCII 码来判别其类型。由 ASCII 码表可知 ASCII 值有以下规律:

ASCII 值小于 32 的为控制字符;

在‘0’和‘9’之间的为数字;

在‘A’和‘Z’之间的为大写字母;

在‘a’和‘z’之间的为小写字母;

其余则为其他字符。

程序如下:

```
#include <stdio.h>
main()
{char c;
printf("请输入一个字符");
c=getchar();
if (c<32)
    printf("这是一个控制字符\n");
else if (c>='0'&&c<='9')
    printf("这是一个数字\n");
}
```

```
else if(c>='A'&&c<='Z')
    printf("这是一个大写字母\n");
else if(c>='a'&&c<='z')
    printf("这是一个小写字母\n");
else
    printf("这是一个其他字符\n");
}
```

第1次运行结果:

请输入一个字符 G
这是一个大写字母

第2次运行结果:

请输入一个字符 (仅输入回车键)
这是一个控制字符

第3次运行结果:

请输入一个字符= (输入等号和回车键)
这是一个其他字符

第4次运行结果:

请输入一个字符 8
这是一个数字

第5次运行结果:

请输入一个字符 b
这是一个小写字母

例 2.17 计算器程序, 从键盘输入一个算术表达式, 输出计算结果。

程序如下:

```
#include <stdio.h>
main()
{float a,b;
 char c;
 printf("请输入一个算术表达式\n");
 scanf("%f%c%f",&a,&c,&b);
 switch(c)
 {case '+': printf("%f\n",a+b); break;
 case '-': printf("%f\n",a-b); break;
 case '*': printf("%f\n",a*b); break;
 case '/': if (b!=0) printf("%f\n",a/b); /*除数不是0则计算出商*/
           else {printf("错误, 输入的除数为0\n"); return (0);}
           break;
 default: printf("输入了错误的运算符\n");
 }
}
```

第1次运行结果:

请输入一个算术表达式
99.3/3
33.100001

第 2 次运行结果:

请输入一个算术表达式
99.3/0
错误,输入的除数为 0

第 3 次运行结果:

请输入一个算术表达式
95+86
181.000000

第 4 次运行结果:

请输入一个算术表达式
95#86
输入了错误的运算符

2.4.3 实验项目

1. 实验目的和要求

- (1) 学会正确使用关系运算符、逻辑运算符及其表达式。
- (2) 熟练掌握 if 语句的使用。
- (3) 熟练掌握 switch 语句的使用。
- (4) 结合程序掌握一些简单的算法。
- (5) 学习调试程序的方法和技巧。

2. 实验内容和步骤

(1) 编程从键盘输入三条边长,判断其是否能构成三角形。若能构成三角形,则判断是否能构成直角三角形,并且只求直角三角形的面积。要求保留 2 位小数。可参考本章范例程序例 2.12。

输入不同的边长分别运行程序 3 次:

能构成三角形,但不是直角三角形。
不能构成三角形。
能构成直角三角形。

(2) 某市实行阶梯水价,居民用户按年度用水量计算,全年用水量划分为三挡,水价逐挡递增,按挡计收:

第一阶梯用水量不超过 144 (含) 立方米,水价为每立方米 4.2 元;
第二阶梯用水量为 144~288 (含) 立方米,水价为每立方米 5.6 元;
第三阶梯用水量为 288 立方米以上,水价为每立方米 9.8 元。

编程从键盘输入一户的全年用水量,计算应缴水费。要求保留 2 位小数。运行程序计算以下 3 个用户的应缴水费。可参考本章范例程序例 2.13。

用户	全年用水量 (立方米)
用户 301	142
用户 302	268
用户 303	306

(3) 某市中考体育考试项目之一是立定跳远，考试的实际测试成绩换算成体育考试分数的标准如下：

男	女	分数
2.41 米以上	1.91 米以上	10 分
2.21~2.40 米	1.73~1.90 米	8 分
1.93~2.20 米	1.49~1.72 米	6 分
1.73~1.92 米	1.33~1.48 米	4 分
1.72 米以下	1.32 米以下	2 分

编程从键盘输入一个考生的性别和实际测试成绩，输出其对应的考试分数。性别可用‘m’表示男，‘w’表示女。可参考本章范例程序例 2.14。

输入以下数据，记录输出结果。

性别	实测成绩（米）	输出结果
m	2.35	
m	1.85	
w	1.93	
w	1.57	
w	-1.83	

(4) 某市信息技术课考试成绩满分为 150 分，考试成绩和对应的考试等级如下：

考试成绩	等级
140 分≤考试成绩≤150 分	优秀
120 分≤考试成绩<140 分	良好
110 分≤考试成绩<120 分	中
90 分≤考试成绩<110 分	及格
考试成绩<90 分	不及格

用 switch 语句编程，从键盘输入一个考生的成绩，输出其对应的等级。可参考本章范例程序例 2.15。

3. 实验思考和总结

- (1) 两种多分支选择 if 语句和 switch 语句有哪些不同？
- (2) 举例说明在求中考体育分数的程序中，需要哪些测试数据才能把程序中的每一个分支都执行一次。
- (3) 总结编程调试的经验教训。

2.5 循环结构程序设计

2.5.1 同步自测

1. 关于下列程序段，以下描述中正确的是（ ）。

```
int k=2;
while (k=0) {printf("%d",k); k--;}
```

- A. while 循环执行 10 次 B. 循环是无限循环
C. 循环体语句一次也不执行 D. 循环体语句执行一次
2. 以下说法正确的是 ()。
- A. 三种循环语句只有 do-while 循环是至少执行一次循环体
B. 语句 for (i=1; i<=10; i++); sum=sum+i; 循环 10 次
C. 循环体不能是多个语句
D. while(表达式)循环体 其中表达式只能是整数
3. 下列程序的输出结果是 ()。

```
#include <stdio.h>
main()
{int j;
 long m=1;
 for (j=1; j<=10; j++)
     m=m*j;
 printf("m=%ld \n" , m) ;
}
```

- A. 11 的阶乘 B. 10 的阶乘 C. 9 的阶乘 D. 无限循环
4. 下列程序段的输出结果是 ()。

```
x=-1;
do
    { x=x*x; }
while (!x);
```

- A. 是死循环 B. 循环执行 2 次
C. 循环执行 1 次 D. 有语法错误
5. 以下 for 循环的执行次数是 ()。

```
for (x=0,y=0; (y=123) && (x<4) ; x++) printf("%d",x);
```

- A. 无限循环 B. 循环次数不定 C. 4 次 D. 3 次
6. 以下程序中, while 循环的循环次数是 ()。

```
#include <stdio.h>
main()
{ int i=0;
 while(i<10)
     {if (i<1) continue;
      if(i==5) break;
      i++;
     }
}
```

- A. 1 B. 10 C. 6 D. 死循环, 不能确定次数

7. 以下程序的输出结果是 ()。

```
#include <stdio.h>
main()
{int a=0,i;
 for(i=1;i<5;i++)
 switch(i)
 {case 0:
 case 3: a+=2;
 case 1:
 case 2: a+=3;
 default: a+=5;
 }
 printf("%d\n",a);
}
```

A. 31 B. 13 C. 10 D. 20

8. 以下程序段的输出结果是 ()。

```
int i,j,m=0;
for (i=1; i<=15; i+=4)
for (j=3; j<=19; j+=4)
m++;
printf("%d\n",m);
```

A. 12 B. 15 C. 20 D. 25

9. 对下列程序段描述正确的是 ()。

```
int x=0,s=0;
while (!x!=0) s+=++x;
printf("%d",s);
```

- A. 运行程序段后输出 0
- B. 运行程序段后输出 1
- C. 程序段中的控制表达式是非法的
- D. 程序段循环无数次

10. 以下程序运行后的输出结果是 ()。

```
#include <stdio.h>
main()
{ int i=10,j=0;
 do
 { j=j+1;
 i--;
 } while (i>2);
 printf("%d\n",j);
}
```

A. 50 B. 52 C. 51 D. 8

11. 程序填空题：以下程序从键盘输入一个 100~999 之间的整数，求出其相应的个位数、十位数和百位数。

```
#include <stdio.h>
main()
{int x,a,b,c;
 do
 {printf("输入一个 3 位数的整数: ");
 scanf("%d", 【1】);
 printf("\n");
 } while (x<100||x>1000);
 a=x/100;
 b=(x-a*100)/10;
 c=x%10 ;
 printf("%d 的个位数是%d, 十位数是%d, 百位数是%d\n", 【2】);
}
```

12. 程序填空题：编写程序求 $1-3+5-7+\cdots-99+101$ 的值。

```
#include <stdio.h>
main()
{ int i, s=1, k=-1;
 for (i=1; 【1】; i++)
 {s=s+k*(【2】);
 k=-k;
 }
 printf("s=%d\n",s);
}
```

13. 程序填空题：下列程序的功能是计算 1~100 之间的整数中，能被 7 整除的那些数之和。

```
#include <stdio.h>
main()
{int i, sum=0;
 for(i=1; i<=100; i++)
 if (【1】%【2】==0) 【3】+=i;
 printf("sum=%d\n",sum);
}
```

14. 填空题：以下程序运行后的输出结果是 ()。

```
#include <stdio.h>
main()
{int n=10;
 while (n>7) {n--; printf("%d",n);}
}
```

15. 填空题：以下程序运行后的输出结果是 ()。

```
#include <stdio.h>
```



```
main()
{int k=4, n=0;
  for (;n<k;)
  {n++;
   if (n%2==0) continue;
   k--;
  }
  printf("%d%d\n",k,n); }
```

16. 判断题：程序有死循环的时候，上机编译不能通过。()
17. 判断题：所有类型的循环都可以进行嵌套使用。()
18. 判断题：语句 `while (5) a++;` 则循环一次也不执行。()
19. 判断题：`break` 语句和 `continue` 语句不可以同时出现在同一个循环体内。()
20. 判断题：利用 `break` 语句可以提前终止循环的执行。()

2.5.2 范例程序解析

例 2.18 求数列前 20 项之和： $\frac{1}{2}, \frac{1}{3}, \frac{2}{4}, \frac{2}{6}, \frac{4}{8}, \dots$

要求累加和保留 2 位小数。

程序如下：

```
#include <stdio.h>
main()
{int i=1;
 float a=1, b=2, c, sum=0, term;
 for (i=1; i<=20; i++)
 {term=a/b;
  sum=sum+term;
  c=a;
  a=b-a; /*求下一个加数的分子*/
  b=c+b; /*求下一个加数的分母*/
 }
 printf("sum=%.2f\n", sum);
}
```

运行结果：

```
sum=8.33
```

思考：(1) 变量 `a`, `b` 改为 `int` 型数据类型，程序运行结果正确吗？为什么？

(2) 若把数列中每一项都以分数的形式显示出来，程序应如何修改？

例 2.19 求和 $1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{6} + \frac{1}{8} - \dots$ 直到某一项的绝对值小于 10^{-5} 为止，该项不累加。要求

累加和保留 8 位小数。


程序如下：

```
#include <stdio.h>
#include <math.h>
main()
{int sign=1; /*sing 表示加数的正负号*/
```

```
double sum=0, n=0, term=1;
while (fabs(term)>=1e-5)
{
    sum=sum+term;
    n=n+2;
    sign=-sign;          /*下一个加数的符号与当前加数相反*/
    term=sign/n;         /*求下一个加数*/
}
printf("sum=%.8f\n" ,sum);
}
```

运行结果:

sum=0.65343141



第1项	1
第2项	-1/2
第3项	1/4
第4项	-1/6
第5项	1/8
第6项	-1/10
第7项	1/12
第8项	-1/14
第9项	1/16
第10项	-1/18
sum=	0.65343141

思考: (1) 本例程和例 2.18 中变量 sum 都用于存放累加和, 为什么定义了不同的数据类型?

图 2-2 例 2.19 修改后的运行结果

(2) 若输出结果如图 2-2 所示, 以上程序应如何修改?

例 2.20 计算多项式前 n 项的值: $sum=a+aa+aaa+aaaa+\cdots$, 从键盘输入 a 为 1~9 之间的整数。例: a=5 n=3 sum=5+55+555

程序如下:

```
#include <stdio.h>
main()
{
    int a,n,i;
    double sum, term;
    printf("请输入 a n 的值");
    scanf("%d%d",&a,&n);
    sum=0; term=a;
    for (i=1; i<=n; i++)
    {
        sum=sum+term;
        term=term*10+a; /*下一个加数*/
    }
    printf("sum=%.0f \n",sum);
}
```

运行结果:

请输入 a n 的值 8 11
sum=98765432088

思考: 将以上程序 double sum, term; 改为 int sum, term;; printf("sum=%.0f\n",sum); 改为 printf("sum=%d\n",sum);, 则 sum 的值显示为负数, 为什么?

例 2.21 求 100~999 之间的整数中, 能被 3 整除且该数的个位或十位或百位含有 5 的三位数。例如 165, 252, 516 等。要求每行显示 10 个数。

程序如下:

```
#include <stdio.h>
main()
{
    int k,a,b,c,i=0;
    printf("能被 3 整除且含有 5 的三位数如下: \n");
}
```

```

for (k=100; k<1000; k++)
if (k%3==0)
{
a=k/100; /*求百位数*/
b=(k/10)%10; /*求十位数*/
c=k%10; /*求个位数*/
if (a==5||b==5||c==5)
{
printf("%6d", k); /*每行显示不满 10 个数则不换行*/
i++;
if (i%10==0) printf("\n"); /*每行显示已满 10 个数则换行*/
}
}
printf("\n");
}

```

运行结果如图 2-3 所示。

能被3整除且含有5的三位数如下:

105	135	150	153	156	159	165	195	225	252
255	258	285	315	345	351	354	357	375	405
435	450	453	456	459	465	495	501	504	507
510	513	516	519	522	525	528	531	534	537
540	543	546	549	552	555	558	561	564	567
570	573	576	579	582	585	588	591	594	597
615	645	651	654	657	675	705	735	750	753
756	759	765	795	825	852	855	858	885	915
945	951	954	957	975					

图 2-3 例 2.21 运行结果

思考: (1) 求 a, b, c 的值还有其它方法吗?

(2) 如果不使用 if (k%3==0) 判断是否能被 3 整除, for 语句如何修改能达到同样的功能?

例 2.22 求 100~300 之间的素数, 每行显示 10 个素数。

程序如下:

```

#include <stdio.h>
#include <math.h>
main()
{
int m, k, i, n=0;
for(m=101; m<=300; m=m+2)
{
k=sqrt(m); /*求m的平方根 k*/
for(i=2; i<=k; i++)
if(m%i==0) break; /*m不是素数,跳出内循环*/
if(i>k) /*m是素数*/
{
printf("%d", m);
n++;
if(n%10==0) printf("\n"); /*每行显示 10 个素数则换行*/
}
}
printf("\n");
}

```

运行结果如图 2-4 所示。

101	103	107	109	113	127	131	137	139	149
151	157	163	167	173	179	181	191	193	197
199	211	223	227	229	233	239	241	251	257
263	269	271	277	281	283	293			

图 2-4 例 2.22 运行结果

思考：若还要求这些素数之和，以上程序应如何修改？

例 2.23 用“穷举法”解决“百元买百鸡”问题。

公鸡每只 5 元，母鸡每只 3 元，小鸡 3 只 1 元。用 100 元买 100 只鸡，问公鸡、母鸡、小鸡各多少只？

程序如下：

```
#include <stdio.h>
main()
{int x, y, z; /*x, y, z 依次表示公鸡、母鸡、小鸡的只数*/
printf("%8s%8s%8s\n", "公鸡","母鸡","小鸡");
for (x=1; x<20; x++) /*公鸡数小于 20 只*/
for (y=1; y<34 ; y++) /*母鸡数小于 34 只*/
{z=100-x-y; /*求小鸡数已满足百鸡条件*/
if (5*x+3*y+z/3.0==100) /*满足百元条件*/
printf("%8d%8d%8d\n", x, y, z);
}
}
```

运行结果：

公鸡	母鸡	小鸡
4	18	78
8	11	81
12	4	84

注意：若将 if (5*x+3*y+z/3.0==100) 中 z/3.0 改为 z/3，则多出 3 行错误的输出，按照题意，小鸡数应能被 3 整除。运行错误结果如下：

公鸡	母鸡	小鸡
3	20	77
4	18	78
7	13	80
8	11	81
11	6	83
12	4	84

以上百元买百鸡问题也可以设计一个三重循环来实现。

程序如下：

```
#include <stdio.h>
main()
{int x, y, z; /*x, y, z 依次表示公鸡、母鸡、小鸡的只数*/
printf("%8s%8s%8s\n", "公鸡","母鸡","小鸡");
```

```
for (x=1; x<20; x++)          /*公鸡数小于 20 只*/
    for (y=1; y<34 ; y++)      /*母鸡数小于 34 只*/
        for (z=1; z<100; z++)  /*小鸡数小于 100 只*/
            if ((5*x+3*y+z/3.0==100)&&(x+y+z==100)) /*满足百元和百鸡两个条件*/
                printf("%8d%8d%8d\n", x, y, z);
}
```

例 2.24 设计一个循环程序，某省居民阶梯电价分为三挡：

第一挡电量：每户年用电量 ≤ 2520 度，收费每度 0.55 元；

第二挡电量：2520 度 $<$ 每户年用电量 ≤ 4800 度，收费每度 0.60 元；

第三挡电量：每户年用电量 > 4800 度，收费每度 0.85 元。

编程从键盘输入一户的全年用电量，计算应缴电费。当输入的全年用电量为-1 时结束循环。要求保留 2 位小数。

程序如下：

```
#include <stdio.h>
main()
{float number,money;
while (1)
    {printf("\n 请输入一户的全年用电量，若输入-1 则结束循环程序");
    scanf("%f" , &number);
    if (number==-1) break; /*输入的用电量为-1,跳出循环，结束运行*/
    if (number<=2520) money=number*0.55; /*年用电量 $\leq 2520$ 度*/
    else if (number<=4800) money=2520*0.55+(number-2520)*0.6;
    else money=2520*0.55+(4800-2520)*0.6+(number-4800)*0.85; /*年用电量
                                                                >4800 度*/

    printf("用电量%.2f 度    电费%.2f 元\n" ,number,money);
    }
}
```

运行结果：

请输入一户的全年用电量，若输入-1 则结束循环程序 2302
用电量 2302.00 度 电费 1266.10 元

请输入一户的全年用电量，若输入-1 则结束循环程序 4507
用电量 4507.00 度 电费 2578.20 元

请输入一户的全年用电量，若输入-1 则结束循环程序 5328
用电量 5328.00 度 电费 3202.80 元

请输入一户的全年用电量，若输入-1 则结束循环程序-1 （输入-1 时结束循环程序）

注意：这是一个循环次数不确定的循环程序，while(1)表示无限循环，本程序是使用循环体中的 if(number== -1) break; 语句跳出循环。既然循环次数是由 number 的值是否为-1 控制的，许多初学者会设计以下错误的程序：

```
#include <stdio.h>
main()
```

```
{float  number=1,money;
while  (number!=-1)
{printf("\n 请输入一户的全年用电量，若输入-1 则结束循环程序");
scanf("%f" , &number);
if (number<=2520) money=number*0.55; /*年用电量≤2520 度*/
else if (number<=4800) money=2520*0.55+(number-2520)*0.6;
else money=2520*0.55+(4800-2520)*0.6+(number-4800)*0.85; /*年用电量
                                                                >4800 度*/

printf("用电量%.2f 度    电费%.2f 元\n" ,number,money);
}
}
```

运行这个错误的程序，如图 2-5 所示，当输入-1 时会输出错误的结果。

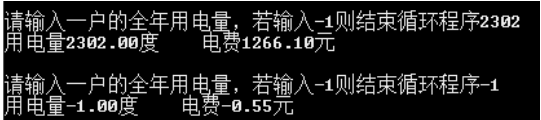


图 2-5 错误程序的运行结果

例 2.25 设计一个循环程序，某公司销售人员的实发工资由底薪 1000 元和销售额提成两部分组成。具体提成比例如下：

销售额	提成比例
销售额≤3000 元	无提成
3000 元<销售额≤6000 元	7%
6000 元<销售额≤10 000 元	10%
销售额>10 000 元	13%

已知销售额单位为元，是一个整数。用 switch 语句编程，从键盘输入一个销售额，计算该员工的实发工资。当输入的销售额为-1 时结束循环。要求实发工资保留 2 位小数。

程序如下：

```
#include <stdio.h>
main()
{int n,sales;
float salary=1000;
while (1)
{printf("请输入一个员工月销售额：");
scanf("%d",&sales);
if (sales== -1) break; /*输入结束标志-1 时，跳出循环*/
if (sales<0) {printf("输入的销售额是负数，请重新输入\n"); continue;}
if (sales%1000==0) /*判断销售额/1000 是否为整数*/
n=sales/1000; /*销售额/1000 是整数时，用此公式求 n 的值*/
else
n=sales/1000+1; /*销售额/1000 不是整数时，用此公式求 n 的值*/
switch(n)
{case 0:
case 1:
```

```

case 2:
case 3: salary = salary +0; break; /*销售额≤3000 元无提成时的实发工资*/
case 4:
case 5:
case 6:
    salary=salary+sales*0.07; /*3000 元<销售额≤6000 元时的实发工资*/
    break;
case 7:
case 8:
case 9:
case 10:
    salary+=sales*0.10;          /*6000 元<销售额≤10000 元时的实发工资*/
    break;
default:
    salary=salary+sales*0.13;     /*销售额>10000 元时的实发工资*/
    break;
}
printf("该员工实发工资为: %.2f 元\n\n",salary); /*输出员工的实发工资*/
}
}

```

运行结果:

请输入一个员工月销售额: 3560

该员工实发工资为: 1249.20 元

请输入一个员工月销售额: 9673

该员工实发工资为: 2216.50 元

请输入一个员工月销售额: -5000

输入的销售额是负数, 请重新输入

请输入一个员工月销售额: 5000

该员工实发工资为: 2566.50 元

请输入一个员工月销售额: -1

例 2.26 设计一个循环程序, 在屏幕最左边输出如图 2-6 所示的图形。

程序如下:

```

#include <stdio.h>
main()
{int i,j,k;
for(i=1; i<=5; i++)
{for(j=1; j<=5-i; j++)
    printf(" ");
for(k=1; k<=2*i-1; k++)
    printf("*");
printf("\n");
}
}

```

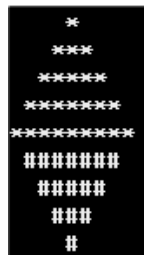


图 2-6 例 2.26 程序输出的图形

```
for(i=1; i<=4; i++)
{for(j=1; j<=i; j++)
    printf(" ");
    for(k=1; k<=9-2*i; k++)
        printf("#");
    printf("\n");
}
```

思考：若在屏幕中间输出如图 2-6 所示的图形，程序应如何修改？

例 2.27 设计一个循环程序，从键盘输入一个整数 n 的值，求 n 的阶乘 $n!=1*2*3*\dots*n$ 。程序如下：

```
#include <stdio.h>
main()
{int i, n;
 double t=1; /*t 存放 n 的阶乘*/
 printf("请输入一个整数");
 scanf("%d", &n);
 for (i=1; i<=n; i++)
     t=t*i; /*循环 n 次累乘得到 n!*/
 printf("%d!=%.0f\n", n, t);
}
```

运行结果：

```
请输入一个整数 4
4!=24
请输入一个整数 12
12!=479001600
```

2.5.3 实验项目

1. 实验目的和要求

- (1) 掌握 while、do...while、for 语句的语法规则、执行流程。
- (2) 掌握利用 while、do...while、for 语句实现循环程序设计的方法。
- (3) 了解 3 种循环语句的异同。

2. 实验内容和步骤

- (1) 求数列前 20 项之和： $\frac{1}{2}, \frac{2}{3}, \frac{3}{5}, \frac{5}{8}, \dots$ 累加和，保留 2 位小数。可参考本章范例程序

例 2.18。

记录运行结果，复制一份已调试成功的源程序，然后修改为循环次数未知的程序。只计算分子、分母之和小于 50 的那些数列各项之和。可参考本章范例程序例 2.19。

(2) 求 100~999 之间的整数中所有的水仙花数。一个水仙花数其各位数字立方和等于该数本身。例如 $153=1^3+5^3+3^3$ ，可参考本章范例程序例 2.21。

(3) 编程从键盘上输入一个整数判断它是否为一个素数，若是素数则求它的阶乘，若不是素数则求它的平方根，要求保留 2 位小数。可参考本章范例程序例 2.22 和例 2.27。

(4) 把 1 张 100 元的人民币兑换成 5 元、2 元和 1 元的纸币（每种都要有）共 50 张，问有哪几种兑换方案？可参考本章范例程序例 2.23。

(5) 36 块砖用 36 人搬，男的一次搬 4 块砖，女的一次搬 3 块砖，小孩 2 人搬 1 块砖。要求一次全搬完，问男、女、小孩各多少人？可参考本章范例程序例 2.23。

(6) 设计一个循环程序，某市实行阶梯水价，居民用户按年度用水量计算，全年用水量划分为三挡，水价逐挡递增，按挡计收：

- 第一阶梯用水量不超过 144（含）立方米，水价为每立方米 4.2 元；
- 第二阶梯用水量为 144~288（含）立方米，水价为每立方米 5.6 元；
- 第三阶梯用水量为 288 立方米以上，水价为每立方米 9.8 元。

编程从键盘输入一户的全年用水量，计算应缴水费，当输入的全年用水量为-1 时结束循环。要求保留 2 位小数。可参考本章范例程序例 2.24。

(7) 设计一个循环程序，某市信息技术课考试成绩满分为 150 分，考试成绩和对应的考等等级如下：

考试成绩	等级
140 分≤考试成绩≤150 分	优秀
120 分≤考试成绩<140 分	良好
110 分≤考试成绩<120 分	中
90 分≤考试成绩<110 分	及格
考试成绩<90 分	不及格

用 switch 语句编程，从键盘输入一个考生的成绩，输出其对应的等级，当输入的成绩为-1 时结束循环。可参考本章范例程序例 2.25。

(8) 设计一个循环程序，在屏幕最左边输出如图 2-7 所示的图形。可参考本章范例程序例 2.26。

思考：以上程序只需要修改其中的一条语句就可以输出如图 2-8 所示的图形，应如何修改程序？



图 2-7 输出 10 行*组成的三角形

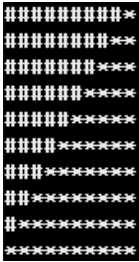


图 2-8 修改程序后输出的矩形

3. 实验思考和总结

- (1) break 语句和 continue 语句有何区别？
- (2) 对循环次数已知的循环通常用哪种循环语句实现？
- (3) 对循环次数不确定的循环通常用哪种循环语句实现？
- (4) 总结编程调试的经验教训。

7. 若有以下说明, 则数值为 5 的表达式是 ()。

```
int a[10]={1,2,3,4,5,6,7,8,9,10};
char c='a',d,g;
```

- A. a[g-c] B. a[4] C. a['d'-'c'] D. a['e'-'c']

8. 下列程序的输出结果是 ()。

```
#include <stdio.h>
#define MAX 10
main()
{int i, sum, a[ ]={1,2,3,4,5,6,7,8,9,10};
 sum=1;
 for(i=0; i<MAX; i++) sum=sum-a[i];
 printf("sum=%d\n",sum);
}
```

- A. sum=55 B. sum=-54 C. sum=-55 D. sum=54

9. 下列程序的输出结果是 ()。

```
#include <stdio.h>
main()
{int y=18, i=0, j, a[8];
 do { a[i]=y%2; i++; y=y/2; } while(y>=1);
 for(j=i-1; j>0; j--) printf("%d",a[j]);
}
```

- A. 1000 B. 1010 C. 1001 D. 1100

10. 下列程序的输出结果是 ()。

```
#include <stdio.h>
main()
{int i, a[ ]={1,2,3,4,5,6,7,8,9};
 for(i=1; i<9; i+=2)
 printf("%d",a[i]);
}
```

11. 程序填空题：将一维数组的 20 个元素按起泡法由小到大排序。

```
#include <stdio.h>
main()
{int a[20], i, j, t;
 printf("请输入 20 个整数:\n");
 for (i=0; i<20; i++)
 scanf("%d", &a[i]);
 for(j=0; j<【1】; j++)
 for(i=0 ;i<【2】 ; i++)
 if (a[i]>a[i+1]) {t=a[i]; a[i]=a[i+1] ; 【3】;}
 printf("排序后各数组元素如下:\n");
 for(i=0; i<20; i++)
 printf("%d ",a[i]);
 printf("\n");
}
```

12. 程序填空题：求数组元素中值大于等于 60 的那些元素的平均值。

```
#include <stdio.h>
main()
{int s, k, i, a[10]={50,60,70,80,90,100,66,44,55,77};
 s=0; k=0;
 for(i=0; i<10; i++)
     if(【1】)
     {s=【2】
      k++; }
     s=【3】
 printf("%d\n",s);
}
```

13. 程序填空题：下列程序中一维数组中共有 8 个元素，求其中最大的元素及该元素的下标。

```
#include <stdio.h>
main()
{int a[8], max, i, k;
 for (i=0; i<8; i++)
     scanf("%d" , &a[i]);
 max=a[0]; k=0;
 for( i=1 ; i<8 ;i++)
     if (【1】)
     {max=a[i];
      【2】 }
 printf("最大元素的值%d 它的下标值%d\n" , 【3】);
}
```

14. 填空题：下列程序的输出结果是（ ）。

```
#include <stdio.h>
main()
{int i,a[10];
 for(i=9; i>=0; i--) a[i]=10-i;
 printf("%d%d%d\n",a[2],a[5],a[8]);
}
```

15. 填空题：下列程序的输出结果是（ ）。

```
#include <stdio.h>
main()
{char ch[7]="678abc36";
 int i, s=0;
 for(i=0; ch[i]!='0'&&ch[i]<='9'; i+=2) s=10*s+ch[i]-'0';
 printf("%d\n",s);
}
```

16. 填空题：下列程序的输出结果是（ ）。

```
#include <stdio.h>
```

```
main()
{char s[ ]="computer";
 s[6]='\0';
 printf("%s\n",s);
}
```

17. 填空题：下列程序的输出结果是（ ）。

```
#include <stdio.h>
main()
{char s1[20]="apple";
 char s2[15]="watermelon";
 printf("%d\n",strlen(strcpy(s2,s1)));
}
```

18. 填空题：下列程序的输出结果是（ ）。

```
#include <string.h>
#include <stdio.h>
main()
{char str[ ]="\x69\072\n";
 printf("%d\n", strlen(str));
}
```

19. 判断题：strcmp(s1,s2)的返回值为一个负数，则表明字符串 s1 一定小于字符串 s2。()

20. 判断题：使用 strlen 函数可以求出一个字符串的实际长度（包含 ‘\0’ 字符）。()

2.6.2 范例程序解析

例 2.28 编程从键盘输入 10 个学生的高考成绩，求高考成绩的最高分和最低分。

程序如下：

```
#include <stdio.h>
#define N 10 /*N 符号常量表示学生人数*/
main()
{int i , a[N] , max, min ;
 for (i=0; i<N; i++)
 {printf("请输入第%d 个同学的分", i+1 );
 scanf("%d" , &a[i]); } /*输入 10 个学生的高考成绩存放到数组 a 中*/
max=min=a[0];
 for (i=1; i<N; i++)
 {if (a[i]>max) max=a[i]; /*max 存放最高分*/
 if (a[i]<min) min=a[i]; /*min 存放最低分*/
 }
 printf("高考最高分: %d 高考最低分: %d \n " ,max , min);
}
```

运行结果：

```
请输入第 1 个同学的分 579
请输入第 2 个同学的分 612
```

请输入第 3 个同学的分 575
请输入第 4 个同学的分 588
请输入第 5 个同学的分 653
请输入第 6 个同学的分 605
请输入第 7 个同学的分 566
请输入第 8 个同学的分 632
请输入第 9 个同学的分 578
请输入第 10 个同学的分 603
高考最高分: 653 高考最低分: 566

例 2.29 某部门 12 名女性已进行血常规体检, 从键盘输入 12 人的血红蛋白数据, 求 12 人的血红蛋白平均值, 统计血红蛋白低于 110g/L 的人数, 并提示“注意饮食营养均衡”。

程序如下:

```
#include <stdio.h>
#define N 12                                /*N 符号常量表示人数*/
#define min 110
main()
{int i, number=0;
 float a[N], average=0;
 for (i=0; i<N; i++)
 {printf("请输入第%d个人的血红蛋白数据", i+1 );
 scanf("%f", &a[i]); } /*输入 N 个血红蛋白数据存放到数组 a 中*/
 for (i=0; i<N; i++)
 {average=average+a[i];
 if (a[i]<min) number++; /*统计血红蛋白低于 110g/L 的人数*/
 }
 average=average/N;
 printf("血红蛋白平均值%.0f 低于 110g/L 的人数%d\n", average, number);
 if (number>0) printf("注意饮食营养均衡\n");
 }
```

运行结果:

请输入第 1 个人的血红蛋白数据 121
请输入第 2 个人的血红蛋白数据 136
请输入第 3 个人的血红蛋白数据 108
请输入第 4 个人的血红蛋白数据 105
请输入第 5 个人的血红蛋白数据 127
请输入第 6 个人的血红蛋白数据 143
请输入第 7 个人的血红蛋白数据 132
请输入第 8 个人的血红蛋白数据 136
请输入第 9 个人的血红蛋白数据 128
请输入第 10 个人的血红蛋白数据 136
请输入第 11 个人的血红蛋白数据 141
请输入第 12 个人的血红蛋白数据 109
血红蛋白平均值 127 低于 110g/L 的人数 3
注意饮食营养均衡

例 2.30 某班 10 名学生参加汉字打字比赛，每名学生都录入 60 个汉字，记录所用时间（单位：秒），编程从键盘输入 10 个学生的打字时间，用起泡法由快到慢排序，并输出第一名的打字成绩。

程序如下：

```
#include <stdio.h>
#define N 10          /*学生人数*/
main()
{int  a[N], i, j, t;
  printf("请输入%d个学生的打字时间（秒）\n" ,N);
  for (i=0; i<N; i++)
    scanf("%d", &a[i]);
  for(j=0; j<N-1; j++)
    for(i=0 ;i<N-1-j ;i++)
      if (a[i]>a[i+1])
        {t=a[i]; a[i]=a[i+1] ;a[i+1]=t;} /*以上双重循环起泡法由小到大排序*/
  printf("打字速度排序如下：\n");
  for(i=0; i<N; i++)
    printf("%d  ",a[i]);    /*输出已排序的 10 个数组元素*/
  printf("\n\n");
  printf("打字速度最快是%d 分%d 秒录入 60 个汉字\n" , a[0]/60, a[0]%60);
}
```

排序后 $a[0]$ 中存放的是打字速度最快的学生所用的秒数， $a[0]/60$ 折合成分钟， $a[0]\%60$ 折合成秒。

运行结果：

```
请输入 10 个学生的打字时间（秒）
36 101 25 66 69 72 125 103 90 83
打字速度排序如下：
25 36 66 69 72 83 90 101 103 125
打字速度最快是 0 分 25 秒录入 60 个汉字
```

例 2.31 某公司 18 名销售人员的实发工资由底薪 1000 元和销售额提成两部分组成。具体提成比例如下：

销售额	提成比例
销售额 \leq 3000 元	0%
3000 元 $<$ 销售额 \leq 6000 元	7%
6000 元 $<$ 销售额 \leq 10000 元	10%
销售额 $>$ 10000 元	13%

编程统计全体人员符合各种提成比例的人数：要求将 18 名销售人员的销售额输入到一维数组中，统计出各种提成比例的人数存放到另一个一维数组中。

程序如下：

```
#include <stdio.h>
#define N 18 /*N 符号常量表示人数*/
main()
```

```

{int i, n, c[4]={0}, sales[N];
for (i=0; i<N; i++)
{
do
{
printf("请输入第%d个人的销售额", i+1 );
scanf("%d" , &sales[i]); /*输入 18 个人的销售额存放到数组 sales 中*/
} while (sales[i]<=0) ; /*此循环确保输入的销售额为正数*/
}
/*以下循环分别统计符合各种提成比例情况的人数存放到数组 c 中*/
for (i=0; i<N; i++)
{if (sales[i]%1000==0) /*判断销售额/1000 是否为整数*/
n=sales[i]/1000; /*销售额/1000 是整数时, 用此公式求 n 的值*/
else
n=sales[i]/1000+1; /*销售额/1000 不是整数时, 用此公式求 n 的值*/
switch(n)
{case 0:
case 1:
case 2:
case 3: c[0]++; break; /*统计销售额≤3000 元, 提成比例为 0%的人数*/
case 4:
case 5:
case 6: c[1]++; break; /*统计 3000 元<销售额≤6000 元, 提成比例为 7%的人数*/
case 7:
case 8:
case 9:
case 10: c[2]++; break; /*6000 元<销售额≤10000 元时, 提成比例为 10%的人数*/
default: c[3]++; /*销售额>10000 元时, 提成比例为 13%的人数*/
}
}
}
/*以下循环输出已统计出的符合各种提成比例情况的人数*/
for (i=0; i<4; i++)
switch(i)
{case 0: printf("提成 0%的人数: %d\n" , c[0]); break;
case 1: printf("提成 7%的人数: %d\n" , c[1]); break;
case 2: printf("提成 10%的人数: %d\n" , c[2]); break;
case 3: printf("提成 13%的人数: %d\n" , c[3]); break;
}
}
}

```

注意: 若在 printf 语句中输出 “%” 要写%%。

运行结果:

```

请输入第 1 个人的销售额 2980
请输入第 2 个人的销售额 3500
请输入第 3 个人的销售额 4600
请输入第 4 个人的销售额 3000
请输入第 5 个人的销售额 5670
请输入第 6 个人的销售额 5800

```


请输入第 7 个人的销售额 2992
请输入第 8 个人的销售额 5128
请输入第 9 个人的销售额 6180
请输入第 10 个人的销售额 12000
请输入第 11 个人的销售额 6290
请输入第 12 个人的销售额 6700

请输入第 13 个人的销售额 6000
请输入第 14 个人的销售额 7685
请输入第 15 个人的销售额 18000
请输入第 16 个人的销售额 8200
请输入第 17 个人的销售额 10000
请输入第 18 个人的销售额 9600
提成 0%的人数: 3
提成 7%的人数: 6
提成 10%的人数: 7
提成 13%的人数:2

例 2.32 将一维数组 a 中 n 个元素重新按逆序存放。

$a[0]$ 与 $a[n-1]$ 交换

$a[1]$ 与 $a[n-2]$ 交换

$a[2]$ 与 $a[n-3]$ 交换

以此类推……

程序如下:

```
#include <stdio.h>
#define N 7 /*N 为数组元素的个数*/
main()
{int i, t, a[N];
 printf("请输入%d个整数", N);
 for (i=0; i<N; i++)
 scanf("%d", &a[i]);
 for (i=0; i<N/2; i++)
 {t=a[i]; /*每次循环完成对应 2 个元素的交换*/
 a[i]=a[N-1-i];
 a[N-1-i]=t;
 }
 for (i=0; i<N; i++)
 printf("%d ",a[i]); /*输出完成逆序存放的数组元素*/
 printf("\n");
}
```

运行结果:

请输入 7 个整数 88 77 99 12 76 125 96
96 125 76 12 99 77 88

例 2.33 删除一个字符串中的指定字符。

程序如下:

```
#include <stdio.h>
#include <string.h>
```

```

/*程序中使用了字符串函数，所以要包含头文件 string.h*/
main()
{char ch1;
 char strg1[50] , temp[50];
 int i, j=0;
 printf("请输入一个字符串 ");
 gets(strg1);
 printf("请输入需删除的字符 ");
 ch1=getchar();
 for (i=0; strg1[i]!='\0'; i++)
     if (strg1[i]!=ch1)
         {temp[j]=strg1[i]; /*不删除的字符存放到 temp 数组中*/
          j++;
         }
 temp[j]='\0'; /*有效字符之后存放字符串结束标志*/
 strcpy(strg1 , temp); /*将 temp 复制给 strg1*/
 puts(strg1); /*输出删除了指定字符之后的字符串*/
}

```

运行结果:

```

请输入一个字符串 what is this?
请输入需删除的字符 h
wat is tis?

```

注意：以上程序中 `temp[j]='\0'` 语句不是可有可无的，若删除该语句，则在输入字符串后程序运行会出错。

例 2.34 编程从键盘输入一行明文（例如英语句子），加密后输出其密文。再选择解密功能将其解密后输出明文。

程序如下：

```

#include <stdio.h>
#include <string.h>
main()
{int i, choose, count=0;
 char a[120], b[120];
 while (1)
 {printf("1 给明文加密，输出密文\n");
  printf("2 给密文解密，输出明文\n");
  printf("3 退出程序\n");
  printf("\n 请选择输入以上数字");
  scanf("%d" , &choose);
  getchar();
  if (choose==1)
  {printf("请输入要加密的明文: \n");
   gets(a); /*从键盘输入明文存入数组 a 中*/
   count=strlen(a); /*求明文有效字符的长度*/
   for(i=0; i<count; i++)
       b[i]=a[i]+i+5; /*将一个明文字符加密存入数组 b 中*/
  }
  }
}

```

```

        b[i]='\0';                /*密文的末尾存入字符串结束标记*/
        printf("加密后的密文是: %s\n",b);    /*输出密文信息*/
    }
    else if (choose==2)
    {count=strlen(a);
    for(i=0; i<count; i++)
        a[i]=b[i]-i-5;          /*将一个密文字符解密, 存入数组 a 中*/
        a[i]='\0';                /*明文的末尾存入字符串结束标记*/
        /*输出明文信息*/
        printf("解密后的明文是: %s\n",a);
    }
    else if (choose==3)
        break; /*跳出循环*/
    else
        printf("输入的数字有错, 请重新输入\n\n");
    }
}

```

思考: 以上程序中的 `gets(a);`, 若输入英语句子时, 可用 `scanf("%s", a);` 替代吗? 为什么? 运行结果如图 2-9 所示。

```

请选择输入以上数字1
请输入要加密的明文:
I am fifty-three years old.
加密后的密文是: N8' iuxxrot 僞>崙叫<78?!枕>?骠喻
1 给明文加密, 输出密文
2 给密文解密, 输出明文
3 退出程序

请选择输入以上数字8
输入的数字有错, 请重新输入

1 给明文加密, 输出密文
2 给密文解密, 输出明文
3 退出程序

请选择输入以上数字2
解密后的明文是: I am fifty-three years old.
1 给明文加密, 输出密文
2 给密文解密, 输出明文
3 退出程序

请选择输入以上数字3

```

图 2-9 例 2.34 程序运行结果

2.6.3 实验项目

1. 实验目的和要求

- (1) 掌握一维数组的基本概念, 定义一维数组和初始化一维数组的方法。
- (2) 掌握一维数组的基本操作, 如输入/输出、引用数组元素等。
- (3) 掌握与使用数组的有关的算法。

2. 实验内容和步骤

- (1) 某企业有高级工程师 15 人, 从键盘输入 15 人的年龄到一维数组中, 求最大年龄和最小年龄。可参考本章范例程序例 2.28。

思考：若高级工程师人数增加到 18 人，程序应如何修改？

(2) 某住宅楼 1 单元有 18 户居民，从键盘输入 18 个家庭的全年用水量到一维数组中，求这个单元的家庭平均年用水量，统计年用水量大于平均值的户数。可参考本章范例程序例 2.29。

(3) 编程实现将 8 名女生的中考立定跳远测试成绩输入到一维数组中，并由高到低按起泡法排序。可参考本章范例程序例 2.30。

(4) 某市信息技术课考试成绩满分为 150 分，考试成绩和对应的考试等级如下：

考试成绩	等级
140 分≤考试成绩≤150 分	优秀
120 分≤考试成绩<140 分	良好
110 分≤考试成绩<120 分	中
90 分≤考试成绩<110 分	及格
考试成绩<90 分	不及格

编程统计全班各考试等级的人数：要求将全班 30 个考生的成绩输入到一维数组中，统计的全班各考试等级的人数存放到另一个一维数组中。可参考本章范例程序例 2.31。

(5) 从键盘输入一串字母存放到一维数组 a 中，然后将该数组逆序存放。可参考本章范例程序例 2.32 和例 2.33。

3. 实验思考和总结

- (1) 解答思考题。
- (2) 总结编程调试的经验教训。

2.7 二 维 数 组

2.7.1 同步自测

- 1. 若定义 `int a[3][4]={0};` 则下列叙述正确的是 ()。
 - A. 只有元素 `a[0][0]` 可得到初值 0
 - B. 此定义不正确
 - C. 数组 a 的各元素可得到初值，但其值不一定为 0
 - D. 数组 a 的每一个元素均可得到初值 0
- 2. 二维数组 `int a[4][4]` 前 5 个元素的存放次序是 ()。
 - A. `a[0][0]` `a[1][1]` `a[2][2]` `a[3][3]` `a[4][4]`
 - B. `a[0][0]` `a[0][1]` `a[0][2]` `a[0][3]` `a[1][0]`
 - C. `a[0][0]` `a[1][0]` `a[2][0]` `a[3][0]` `a[0][1]`
 - D. `a[0]` `a[0][0]` `a[0][0][0]` `a[0][0][0][0]` `a[0][0][0][0][0]`
- 3. 有两个字符数组 a,b 则以下正确的语句是 ()。
 - A. `gets(a,b);`
 - B. `scanf("%s%s",a,b);`
 - C. `scanf("%s%s",&a,&b);`
 - D. `gets("a"), gets("b");`

- 判断字符串 s1 是否大于 s2, 应当使用 ()。
A. if(s1>s2) B. gets("a"),gets("b");
C. if(strcmp(s2,s1)>0) D. if(strcmp(s1,s2)>0)
- 若有定义 int a[3][4];, 则对数组 a 的元素的引用非法的是 ()。
A. a[2][2*1] B. a[1][3] C. a[4-2][0] D. a[0][4]
- 以下不能对二维数组 a 进行正确初始化的语句是 ()。
A. int a[2][3]={0}; B. int a[][3]={ {1,2},{0} };
C. int a[2][3]={ {1,2},{3,4},{5,6} }; D. int a[][3]={1,2,3,4,5,6};
- 下列数组定义语句中, 正确的是 ()。
A. char a[][]={'a','b','c','d','e','f'}; B. char a[2][3]='a','b';
C. char a[][3]='a','b','c','d','e','f'; D. char a[][]={{'a','b','c','d','e','f'}};
- 若以下程序运行时输入: 2 4 6<回车>, 则程序的输出结果是 ()。
A. 2 0 0 B. 2 0 4 C. 2 4 0 D. 2 4 6

```
#include <stdio.h>
main()
{int x[3][2]={0},i;
  for(i=0;i<3;i++) scanf("%d",x[i]);
  printf("%3d%3d%3d\n",x[0][0],x[0][1],x[1][0]);
}
```

9. 下列程序的输出结果是 ()。

```
#include <stdio.h>
main()
{int i;
 int x[4][4]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16};
 for(i=0;i<4;i++) printf("%3d",x[i][3-i]);
}
```

- A. 1 5 9 13 B. 1 6 11 16
C. 4 7 10 13 D. 4 8 12 16
10. 当接受用户输入的含有空格的字符串时，应使用函数（ ）。
- A. gets() B. getchar() C. scanf() D. printf()
11. 填空题：下列程序的输出结果是（ ）。

```
#include <stdio.h>
main()
{int i,j, a[3][4]={1,2,3,4,5,6,7,8,9,10,11,12};
  for (i=0; i<3; i++)
    for(j=0; j<4; j++)
      if(i==2-j) printf("%d",a[i][j]);
}
```

12. 填空题: 下列程序的输出结果是 ()。

```
#include <stdio.h>
main()
```

```
{int i,j, s=0;
int a[3][3]={5,6},{7,8},{9,10}};
for(i=1;i<3;i++)
    for(j=0;j<i;j++)
        s+=a[i][j];
printf("%d",s);
}
```

13. 填空题：下列程序的输出结果是（ ）。

```
#include<stdio.h>
main()
{int i,j, a[2][3]={1,2,3,4,5,6};
for(i=0; i<2; i++)
for(j=2; j>=0; j-=2)
printf("%d",a[i][j]);
}
```

14. 填空题：下列程序的输出结果是（ ）。

```
#include<stdio.h>
main()
{int a[6][6],i,j;
for(i=1;i<6;i++)
for(j=1;j<6;j++)
a[i][j]=(i/j)*(j/i);
for(i=1;i<6;i++)
{for(j=1;j<6;j++)
printf("%2d",a[i][j]);
printf("\n");
}
}
```

15. 填空题：下列程序的输出结果是（ ）。

```
#include <stdio.h>
main()
{int a[2][3]={1,2,3},{4,5,6}}, b[3][2], i, j;
for(i=0; i<2; i++)
for(j=0; j<3; j++)
b[j][i]=a[i][j];
for(i=0; i<3; i++)
{for(j=0; j<2; j++)
printf("%-3d",b[i][j]);
printf("\n"); }
}
```

16. 填空题：下列程序的输出结果是（ ）。

```
#include <stdio.h>
main()
```

```
{int i,j,a[2][3]={4,5,6,1,2,3},b[2][3];
for(i=0;i<2;i++)
    for(j=0;j<3;j++)
        if(j<2) b[i][j+1]=a[i][j] ;
for(i= 0; i< 2;i++)
    b[i][0]=a[i][2];
for(i=0;i<2;i++)
{for(j=0;j<3;j++)
    printf("%d",b[i][j]);
printf("\n");
}
}
```

17. 程序填空题：求二维数组所有元素之和。

```
#include <stdio.h>
main()
{ int a[3][4] , i, j, sum=0;
printf("请输入数组元素值\n");
for (i=0; i<3; i++)
    for (j=0; j<4; j++)
        【1】
for (i=0; i<3; i++)
    for (j=0; j<4; j++)
        sum= 【2】
printf("sum=%d\n" , sum);
}
```

18. 程序填空题：求二维数组包括主对角线元素在内的上三角形那部分元素之和。
(主对角线上的元素行、列下标相等)

```
#include <stdio.h>
#define N 【1】
main()
{int i,j,s=0;
int x[N][N]={ {3,5,1,4},{7,4,5,8},{1,6,5,3},{2,4,6,8}};
for(i=0; i<N; i++)
    for( 【2】 ; j<N; j++)
        s= 【3】
printf("%d\n", s);
}
```

19. 程序填空题：求3个字符串中最大的字符串。

```
#include <stdio.h>
#include <string.h>
main()
{char string[20];
char str[3][20];
int i;
```

```

printf("请输入 3 个字符串\n");
for(i=0;i<3;i++)
gets( 【1】 );
if(strcmp(str[0], str[1])>0)
strcpy(string , str[0]);
else
strcpy (string , str[1]);
if(strcmp( str[2] , string)>0)
strcpy(string, str[2]);
printf("\n 最大字符串:%s\n", 【2】);
}

```

20. 程序填空题：下列程序显示杨辉三角形的前 8 行，运行结果如下：

```

1
1   1
1   2   1
1   3   3   1
1   4   6   4   1
1   5   10  10   5   1
1   6   15  20  15   6   1
1   7   21  35  35  21   7   1

```

程序如下：

```

#include <stdio.h>
#define N 8
main()
{int i, j , a[N][N];
  for (i=0 ; i<N ; i++)
  { a[i][0]=1;
    a[i][i]=1;
  }
  for (i=2; i<N; i++)
    for (j=1; j<i ; j++)
      a[i][j]=【1】
  for (i=0; i<N; i++)
    {for(j=0 ; 【2】 ; j++) printf("%5d" , a[i][j]);
      printf("\n");
    }
}

```

2.7.2 范例程序解析

例 2.35 从键盘输入一个 8 行 4 列的二维数组的各元素值，靠外侧的元素存放的是全体工程师的年龄，求工程师的平均年龄和工程师中年龄最大值。

程序如下：

```

#include <stdio.h>
#define M 8

```

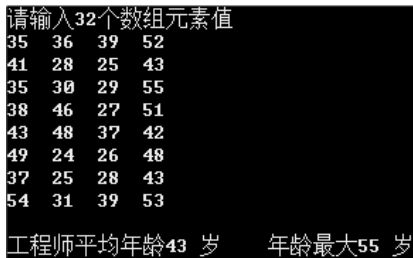


```

#define N 4
main()
{int a[M][N], i, j, average=0 ,max=0 ,number=0;
printf("请输入%d个数组元素值\n" , M*N);
for (i=0; i<M; i++)
    for (j=0; j<N; j++)
        scanf("%d",&a[i][j]);                /*输入 32 个数组元素的值*/
printf("\n");
for (i=0; i<M; i++)
    for (j=0; j<N; j++)
        if (i==0 || i==M-1|| j==0 || j==N-1)
            {average=average+a[i][j];          /*外侧元素即工程师的年龄累加*/
            number++;                          /*统计工程师人数*/
            if (a[i][j]>max) max=a[i][j];        /*求工程师年龄最大值*/
            }
average=average/number;                      /*求工程师年龄平均值*/
printf("工程师平均年龄%d 岁    年龄最大%d 岁\n",average , max);
}

```

运行结果如图 2-10 所示。



```

请输入32个数组元素值
35 36 39 52
41 28 25 43
35 30 29 55
38 46 27 51
43 48 37 42
49 24 26 48
37 25 28 43
54 31 39 53
工程师平均年龄43 岁    年龄最大55 岁

```

图 2-10 例 2.35 程序运行结果

例 2.36 分别求 3×3 矩阵的两条对角线上的元素之和。

程序如下：

```

#include <stdio.h>
#define N 3
main()
{int a[N][N] , i, j, sum1=0, sum2=0;
printf("请输入%d个数组元素的值\n" , N*N);
for (i=0; i<N; i++)
    for (j=0; j<N; j++)
        scanf("%d",&a[i][j]);                /*输入 9 个数组元素的值*/
for (i=0; i<N; i++)
    for (j=0; j<N; j++)
        {if (i==j) sum1=sum1+a[i][j];          /*分别求对角线元素之和*/
        if ((i+j)==N-1) sum2=sum2+a[i][j];
        }
printf(" 两条对角线元素之和分别是: sum1=%d    sum2=%d\n" , sum1,sum2);
}

```

运行结果如图 2-11 所示。

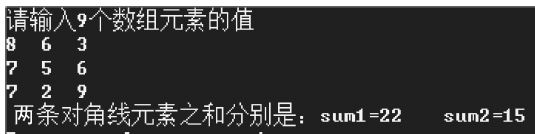


图 2-11 例 2.36 程序运行结果

思考：若修改以上程序中的两个 if 语句，修改后的下列程序计算出的 sum2 的值是错误的，为什么？

错误的运行结果如图 2-12 所示。

```
#include <stdio.h>
#define N 3
main()
{int a[N][N] , i, j, sum1=0, sum2=0;
printf("请输入%d 个数组元素的值\n" , N*N);
for (i=0; i<N; i++)
    for (j=0; j<N; j++)
        scanf("%d",&a[i][j]);          /*输入 9 个数组元素的值*/
for (i=0; i<N; i++)
    for (j=0; j<N; j++)
        {if (i==j) sum1=sum1+a[i][j]; /*分别求对角线元素之和*/
          else if ((i+j)==N-1) sum2=sum2+a[i][j];
        }
printf("两条对角线元素之和分别是: sum1=%d sum2=%d\n" , sum1,sum2);
}
```

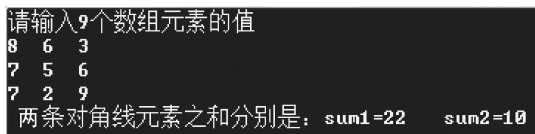


图 2-12 例 2.36 被修改后的程序错误的运行结果

例 2.37 某家电商场 2015 年 12 个月家电营业额（单位：万元）如下：

月份	电视机	电冰箱	空调	洗衣机
1	112.3	85.2	108.0	116.7
2	212.6	79.2	98.6	96.5
3	305.7	113.5	86.2	101.0
4	93.6	201.7	83.6	153.7
5	216.0	320.7	126.5	173.1
6	142.5	215.3	136.8	95.6
7	115.3	185.2	172.5	106.2
8	112.6	195.2	198.6	86.5
9	205.7	173.5	92.6	121.0

10	321.6	281.7	186.3	133.7
11	215.8	220.7	96.5	153.1
12	139.3	115.9	99.6	195.6

从键盘输入以上二维数组 a (不包括月份), 求各种家电的全年平均营业额、最大月营业额、最小月营业额存放到二维数组 b 中。

```
#include <stdio.h>
main()
{int i, j;
 float a[12][4], b[3][4], max, min;
 printf("请输入 12 个月各种家电的营业额\n ");
 for (i=0; i<12; i++)
     for (j=0; j<4; j++)
         scanf("%f", &a[i][j]); /*输入 12 个月的各种营业额*/
 for (j=0; j<4; j++)
 {b[0][j]=0;
  for (i=0; i<12; i++)
      b[0][j]=b[0][j]+a[i][j];
  b[0][j]=b[0][j]/12; /*求各种家电平均营业额存放到数组 b 第 0 行的 4 个元素中*/
 }
 for (j=0; j<4; j++)
 {max=0; min=10000000;
  for (i=0; i<12; i++)
      {if (a[i][j]>max) max=a[i][j];
       if (a[i][j]<min) min=a[i][j];
      }
  b[1][j]=max; /*求各种家电最大营业额存放到数组 b 第 1 行的 4 个元素中*/
  b[2][j]=min; /*求各种家电最小营业额存放到数组 b 第 2 行的 4 个元素中*/
 }
 for (i=0; i<3; i++)
 {switch(i)
 {case 0: printf("\n\n 平均营业额 (万元): \n%-12s%-12s%-12s%-12s\n" , \
 "电视机", "电冰箱", "空调", "洗衣机"); break;
 case 1: printf("\n\n 最大营业额 (万元): \n%-12s%-12s%-12s%-12s\n" , \
 "电视机", "电冰箱", "空调", "洗衣机"); break;
 case 2: printf("\n\n 最小营业额 (万元): \n%-12s%-12s%-12s%-12s\n" , \
 "电视机", "电冰箱", "空调", "洗衣机"); break;
 }
 for (j=0; j<4; j++)
     printf("%-12.2f" , b[i][j]); /*输出数组 b 的元素, 保留 2 位小数*/
 printf("\n");
 }
 }
```

注意: 以上程序中一条 printf 语句分两行写, 第一行要加 “\” 作为行结束标志。运行结果如图 2-13 所示。



图 2-13 例 2.37 程序运行结果

例 2.38 将某教材 unit1 的单词以赋初值的形式存入二维数组中，用起泡法按字母 a~z 的顺序排序。

程序如下：

```
#include <stdio.h>
#include <string.h>
#define M 42
#define N 50
main()
{int i,j;
char t[N];
char c[M][N]={{"honest      adj 诚实的,正直的"}, {"brave      adj 勇敢的"}, \
{"loyal      adj 忠诚的,忠心的"}, {"wise      adj 英明的,明智的,聪明的"}, \
{"handsome  adj 英俊的,大方的,美观的"}, {"smart  adj 聪明的,漂亮的,敏捷的"}, \
{"argue      vt 争论,辩论"}, {"solution  n 解答,解决办法,解决方案"}, \
{"classical  adj 古典的,古典文学的"}, {"fond      adj 爱的,多情的,喜欢的"}, \
{"fond of    喜欢,爱好"}, {"match      n 火柴"}, \
{"match      n 火柴"}, {"mirror      n 镜子"}, \
{"fry        vt&vi 油煎,油炸"}, {"gun      n 炮,枪"}, \
{"hammer     n 锤子,槌"}, {"sawn      &vt&vi 锯"}, \
{"rope       n 绳,索,绳索"}, {"compass   n 罗盘,指南针"}, \
{"movie      n 电影"}, {"cast       vt&vi 投掷,投射,抛"}, \
{"survive    vt 幸免于,从……中生还 vi 幸存"}, {"deserted  adj 荒芜的,荒废的"}, \
{"hunt       vt&vi&n 打猎,猎取,搜寻"}, {"hunt for   搜索,追寻,寻找"}, \
{"in order to 为了"}, {"share      vt&vi 分享,共有,分配 n 共享,份额"}, \
{"sorrow     n 悲哀,悲痛"}, {"care about 担心,关心"}, \
{"feeling    n. 触觉,知觉,感觉,情绪"}, {"such as    例如"}, \
{"airplane   n 飞机"}, {"parachute  n 降落伞"}, \
```

```

{"lie      n 谎话,谎言"}, {"speech    n 演说,讲话,语音"}, \
{"adventure n&vt&vi 冒险,冒险经历"}, {"notebook  n 笔记本,笔记式电脑"}, \
{"e-pal     n 网友"}, {"drop sb a line  给某人写信(通常指写短信)"}, \
{"formal    adj 正式的,正规的"}, {"error      n 错误,差错"};
for(j=0; j<M-1; j++)
    for(i=0 ;i<M-1-j ;i++)
        if (strcmp(c[i],c[i+1])>0)
            {strcpy(t,c[i]); strcpy(c[i],c[i+1]); strcpy(c[i+1],t) ;}
                                                    /*起泡法由小到大排序*/

for(i=0; i<M; i++)
    puts(c[i]);
}

```

注意：将二维字符数组赋初值的一条语句写成多行，除最后一行加分号之外，其余各行应加“\”作为行结束标志。

思考：若单词及词义从键盘上输入，程序应如何修改？

运行结果如图 2-14 所示。

```

adventure  n&vt&vi 冒险,冒险经历
airplane   n 飞机
argue      vt 争论,辩论
brave      adj 勇敢的
care about  担心,关心
cast       vt&vi 投掷,投射,抛
classical  adj 古典的,古典文学的
compass    n 罗盘,指南针
deserted   adj 荒荒的,荒废的
drop sb a line  给某人写信(通常指写短信)
e-pal      n 网友
error      n 错误,差错
feeling    n. 触觉,知觉,感觉,情绪
fond       adj 爱的,多情的,喜欢的
fond of    喜欢,爱好
formal     adj 正式的,正规的
fry        vt&vi 油煎,油炸
gun        n 炮,枪
hammer     n 锤子,槌
handsome   adj 英俊的,大方的,美观的
honest     adj 诚实的,正直的
hunt       vt&vi&n 打猎,猎取,搜寻
hunt for   搜索,追寻,寻找
in order to 为了
lie        n 谎话,谎言
loyal      adj 忠诚的,忠心的
match      n 火柴
match      n 火柴
mirror     n 镜子
movie      n 电影
notebook   n 笔记本,笔记式电脑
parachute  n 降落伞
rope       n 绳,索,绳索
sawn       &vt&vi 锯
share      vt&vi 分享,共有,分配 n 共享,份额
smart      adj 聪明的,漂亮的,敏捷的
solution   n 解答,解决办法,解决方案
sorrow     n 悲哀,悲痛
speech     n 演说,讲话,语音
such as    例如
survive    vt 幸免于,从……中生存 vi 幸存
wise       adj 英明的,明智的,聪明的

```

图 2-14 例 2.38 程序运行结果

2.7.3 实验项目

1. 实验目的和要求

- (1) 掌握字符数组的基本概念，以及定义字符数组和初始化字符数组的方法。
- (2) 掌握字符数组和字符串的关系，以及用字符串初始化字符数组的方法。
- (3) 了解常用字符串处理函数的使用方法。

2. 实验内容和步骤

- (1) 求一个 5×6 矩阵的所有靠外侧的元素之和，元素均为整数。可参考本章范例程序例 2.35。
- (2) 分别求一个 4×4 矩阵的一条对角线上的元素之和以及另一条对角线上的元素之乘积。可参考本章范例程序例 2.36。
- (3) 某学习小组 12 人，每人有 5 门课的考试成绩，从键盘输入 12 人的考试成绩，存放到二维数组 a[12][5]，求 5 门课程的平均分（整数）存入一维数组 b。可参考本章范例程序例 2.37。
- (4) 某大学建筑 151 班 27 名男生住在 2 号宿舍楼 301 室、302 室、303 室、304 室、305 室。名单及住址如下：

姓名	住址
李新宇	2-303
张正毅	2-301
刘建华	2-301
方斌	2-303
武鹏	2-301
高海涛	2-301
李玉锦	2-302
张晓阳	2-303
宋立杉	2-301
赵浩	2-302
许宇	2-305
陈天浩	2-303
孙志刚	2-302
赵军健	2-301
王翔宇	2-304
范涛	2-305
付晨	2-304
王晓旭	2-302
鲁海涛	2-302
郭帅	2-305
王云飞	2-305

陈伟伟	2-304
管青	2-305
孙亮	2-305
钱嘉明	2-304
王鹏飞	2-302
姜山	2-302

编程利用以上学生信息给一个二维数组赋初值,统计每个宿舍的学生人数,并输出按住址由小到大排列的各宿舍学生名单。可参考本章范例程序例 2.38。

提示:将字符串“2-303 李新宇”存入字符数组第 0 行,“2-301 张正毅”存入字符数组第 2 行,以此类推,按起泡法排序。

3. 实验思考和总结

- (1) 解答范例程序或实验项目中的思考题。
- (2) 总结编程调试的经验教训。

2.8 函数和编译预处理

2.8.1 同步自测

1. 以下正确的函数定义是 ()。
A. `double fun(int x,int y)`
 `{ z=x+y; return z; }`
B. `double fun(int x,y)`
 `{ int z; return z; }`
C. `fun (x,y)`
 `{ int x,y; double z;`
 `z=x+y; return z; }`
D. `double fun(int x,int y)`
 `{ double z;`
 `return z; }`
2. 以下正确的说法是 ()。
A. 形参是虚拟的,不占用存储单元
B. 实参与其对应的形参共用一个存储单元
C. 当实参与其对应的形参同名时才共用相同的存储单元
D. 实参与其对应的形参各占用独立的存储单元
3. 若用数组名作为函数调用的实参,则传递给形参的是 ()。
A. 数组中第一个元素的值
B. 数组的首地址
C. 数组元素的个数
D. 数组中的全部元素的值
4. C 语言中形参的默认存储类别是 ()。
A. 自动 (auto)
B. 静态 (static)
C. 寄存器 (register)
D. 外部 (extern)
5. 下列程序存在语法错误,关于错误原因的正确说法是 ()。

```
#include <stdio.h>
main()
```

```
{int  A=5, b;  
void  p_ch();  
...  
b=p_ch(A);  
... }
```

- A. 语句 `void p_ch();` 有错, 它是函数调用语句, 不能使用 `void` 说明
- B. 变量名不能使用大写字母
- C. 函数说明和函数调用语句之间有矛盾
- D. 函数名不能使用下划线

6. 下列程序运行后, 输出结果是 ()。

```
#include <stdio.h>  
#define  ADD(x) x+x  
main()  
{ int m=1, n=2, s=3;  
  s=S+ADD(m+n);  
  printf("s=%d\n", s);  
}
```

- A. s=8 B. s=9 C. s=6 D. s=18

7. 下列程序运行后, 输出结果是 ()。

```
#include <stdio.h>  
int func(int n)  
{int k=1, sum=0,i;  
  for(i=1;i<=n;i++) sum+=(k*=i);  
  return (sum);  
}  
main()  
{printf("func(4)=%d\n", func(4));  
}
```

- A. func(4)=18 B. func(4)=33 C. func(4)=13 D. func(4)=25

8. 下列程序运行后, 输出结果是 ()。

```
#include <stdio.h>  
#define PT printf  
#define S(x) 2.1*x*x  
main()  
{int x=1, y=2;  
  printf("%.1f\n" , S(x+y));}
```

- A. 8.2 B. 程序有错 C. 7.1 D. 6.1

9. 下列程序运行后, 输出结果是 ()。

```
#include <stdio.h>  
int func()  
{static int n=3;
```



```
n+=2;
return(n);
}
main ()
{int i,n;
 for(i=1; i<=5; i++) n=func();
 printf("n=%d\n",n);
}
```

A. n=13

B. n=11

C. n=25

D. n=15

10. 下列程序运行后, 输出结果是 ()。

```
#include<stdio.h>
int f(int x, int y)
{int k=2*x+y;
 return(k);}
main()
{int a1=2, a2=4, a3=6, s;
 s=f(f(a1,a2),f(a2,a3));
 printf("%d\n",s);
}
```

A. 20

B. 32

C. 24

D. 30

11. 填空题: 下列函数调用语句中含有 () 个实参。

```
func((e1,e2),(e3,e4,e5));
```

12. 填空题: 下列程序输出结果是 ()。

```
#include <stdio.h>
int function(int num)
{int k=1;
 do
 {k*=num%10;
 num/=10;
 }
 while(num);
 return (k);
}
main()
{int n=28;
 printf("%d\n",function(n) );
}
```

13. 填空题: 下列程序输出结果是 ()。

```
#include <stdio.h>
void fun(int b[ ], int n)
{ int i, t;
 for(i=0; i<n/2; i++)
 {t=b[i]; b[i]=b[n-1-i]; b[n-1-i]=t;}}
```

```

}
main()
{ int a[10]={1,2,3,4,5,6,7,8,9,10}, i;
  fun(a,10);
  for(i=2; i<8; i++)
    printf("%d", a[i]);
  printf("\n");
}

```

14. 填空题：下列程序输出结果是（ ）。

```

#include <stdio.h>
#define N 3
void fun(int a[ ][N], int b[ ])
{int i;
  for(i=0; i<N; i++) b[i]=a[i][i];
}
main()
{ int x[ ][N]={ {1,2,3}, {4,5,6}, {7,8,9} }, y[N], i;
  fun(x,y);
  for (i=0; i<N; i++) printf("%d,", y[i]);
  printf("\n");
}

```

15. 填空题：若从键盘输入 8,7,9<回车>，下列程序输出结果是（ ）。

```

#include <stdio.h>
void func(int a, int b, int c)
{if(a>b)
  func(c, a, b);
  else printf("%d,%d,%d\n",a,b,c);
}
main()
{int x,y,z;
  scanf("%d,%d,%d",&x, &y, &z);
  func(x, y, z);
}

```

16. 程序填空题：下列程序的功能是调用函数 fun，计算： $m = 1 - 2 + 3 - 4 + \cdots + 9 - 10$ ，并输出结果。

```

int fun(int n)
{int m=0, f, i;
  【1】
  for (i=1; i<=n; i++)
  {m+= 【2】 *f ;
    f=-f ;
  }
}

```

```
    return (【3】);
}
main()
{printf ("m=%d\n" , fun (10 ) ) ;
}
```

17. 程序填空题：下列程序实现的是找出一个 2×4 矩阵中的最大元素值。

```
#include <stdio.h>
int max_value(int arr[ ][4])
{int i, j, max;
max=arr[0][0];
for (i=0; i<2; i++)
    for (j=0; j<【1】; j++)
        if (arr[i][j]>max)
            【2】
return (【3】);
}
main()
{int a[2][4]={1,5,2,99,7,3,6,8};
printf("max=%d\n" , 【4】 );
}
```

18. 程序填空题：下列程序的功能是计算 $1!+2!+3!+4!+5!$ 的阶乘之和。

```
#include <stdio.h>
long func(int n)
{int i ;
long s;
【1】
for(i=1; i<=n; i++) s=【2】
return (s);
}
main()
{long sum;
int k, n=5;
sum=0;
for(k=1; k<=n; k++)
    sum=sum+【3】
printf("%ld\n",sum);
}
```

19. 判断题：函数定义 `void max(int a,int b)` 表示该函数的返回值不确定。()
20. 判断题：在 C 语言自定义函数中，形参可以是变量、常量或表达式。()

2.8.2 范例程序解析

例 2.39 编程利用主函数调用自定义函数，输出如图 2-15 所示的图形，要求屏幕背景为白色，字符颜色为浅蓝色。

```
#include <stdio.h>
#include <stdlib.h>
void prc(char ch, int n)
{int i;
 for (i=1; i<=n; i++)
  printf("%c", ch);
}
```

```

      *
     ***
    *****
   *********
  ***********
 *****
  *****
   *****
    *****
     *****
      *****

```

```
main()
{int i;
 system("cls");
 system("color F9");
 for (i=1; i<=5; i++)
 {prc( ' ' , 5-i);
  prc( '*' , 2*i-1);
  printf("\n"); }
 for (i=4; i>=1; i--)
 {prc( ' ' , 5-i);
  prc( '#' , 2*i-1);
  printf("\n"); }
}
```

图 2-15 例 2.39 程序运行结果

/*清除屏幕显示信息*/
/*设置屏幕背景为白色，字符颜色为蓝色*/
/*输出同一行“*”之前的空格*/
/*输出同一行空格之后的“*”*/
/*输出同一行“#”之前的空格*/
/*输出同一行空格之后的“#”*/

若改为图形从第 5 个制表位开始输出，并且屏幕背景为浅蓝色，字符颜色为浅黄色，则程序如下：

```
#include <stdio.h>
#include <stdlib.h>
void prc(char ch, int n)
{int i;
 for (i=1; i<=n; i++)
  printf("%c", ch);
}

main()
{int i;
 system("cls");
 system("color 9E");
 for (i=1; i<=5; i++)
 {printf("\t\t\t\t\t");
  prc( ' ' , 5-i);
  prc( '*' , 2*i-1);
  printf("\n"); }
 for (i=4; i>=1; i--)
 {printf("\t\t\t\t\t");
  prc( ' ' , 5-i);
  prc( '#' , 2*i-1);
  printf("\n"); }
}
```

/*清除屏幕显示信息*/
/*设置屏幕背景为浅蓝色，字符颜色为浅黄色*/
/*使第一行的“*”从第 5 个制表位开始输出*/
/*输出同一行“*”之前的空格*/
/*输出同一行空格之后的“*”*/
/*输出同一行“#”之前的空格*/
/*输出同一行空格之后的“#”*/

例 2.40 某商业银行一年以上定期存款年利率如下：

存款项目	年利率
1 年定期存款	1.50%
3 年定期存款	2.10%
5 年定期存款	2.75%

编程利用主函数调用自定义函数，求到期时的本息之和。当从键盘输入的存款数为 0 时程序结束。

程序如下：

```
#include <stdio.h>
float func(float deposit, int years)
/*利率 rate, 存款本金 deposit , 存款年数 years, 本金及利息总和 total*/
{float rate, total;
switch (years)
{case 1: rate=0.015; break;
case 3: rate=0.021; break;
case 5: rate=0.0275; break;
}
total=deposit*rate*years;
return (total); /*返回值为利息*/
}

main()
{float total, deposit;
int years;
while (1)
{printf("请输入存款数 年数 (存款数输入 0 结束程序)");
scanf("%f%d", &deposit, &years);
if (deposit==0) break;
if (years==1 || years==3 || years==5)
{total=deposit+func(deposit, years);
printf("存款本金: %.2f 元 年数: %d 本息之和: %.2f 元\n\n\n", deposit, years, total); }
else
printf("输入的年数有错, 请重新输入\n");
}
}
```

运行结果如图 2-16 所示。

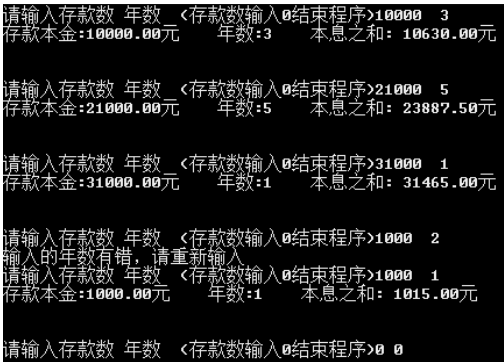


图 2-16 例 2.40 程序运行结果

注意：以上程序若用宏定义 `#define PT printf`，即用宏名 `PT` 表示 `printf`，运行已修改的下列程序也可得到如图 2-16 所示的正确结果。

```
#include <stdio.h>
#define PT printf
float func(float deposit, int years)
/*利率 rate , 存款本金 deposit , 存款年数 years, 本金及利息总和 total*/
{float rate , total;
 switch (years)
 {case 1: rate=0.015; break;
 case 3: rate=0.021; break;
 case 5: rate=0.0275; break;
 }
 total=deposit*rate*years;
 return (total); /*返回值为利息*/
}

main()
{float total, deposit;
 int years;
 while (1)
 {PT("请输入存款数 年数 (存款数输入 0 结束程序)");
 scanf("%f%d", &deposit, &years);
 if (deposit==0) break;
 if (years==1 || years==3 || years==5)
 {total=deposit+func(deposit, years);
 PT("存款本金: %.2f 元 年数: %d 本息之和: %.2f 元\n\n", deposit, years, total); }
 else
 PT("输入的年数有错, 请重新输入\n");
 }
}
```

例 2.41 编程将一维数组作为形式参数：在自定义函数中求一个月 n 个交易日内某只股票的平均收盘价，并用起泡法将每天的收盘价由高到低排序。主函数从键盘输入每天的收盘价存放到一维数组中，并输出平均收盘价及该数组由大到小排序后的值。

程序如下：

```
#include <stdio.h>
#define N 23 /*一个月内交易的天数*/
float sort(float price[ ], int days)
{int i, j;
 float t , aprice=0;
 for(i=0 ; i<days ; i++)
 aprice=aprice+price[i];
 aprice=aprice/days; /*求平均收盘价*/
 /*以下双重循环起泡法实现由高到低排序*/
 for(j=0 ; j<days-1 ; j++)
 for(i=0 ; i<days-1-j ; i++)
 if (price[i]<price[i+1])
```

```

        {t=price[i];
        price[i]=price[i+1];
        price[i+1]=t;
        }
    return (aprice);
}

main()
{int i;
 float price[30], aprice;
 printf("请输入%d 个该股票的收盘价\n" ,N);
 for (i=0; i<N; i++)
     scanf("%f", &price[i]);
 aprice=sort(price, N);
 printf("\n\n 由高到低排序的收盘价如下: \n");
 for(i=0; i<N; i++)
 {printf("%.2f    ",price[i]);          /*输出已排序的 N 个数组元素*/
  if ((i+1)%8==0) printf("\n");        /*每行输出 8 个元素*/
 }
 printf("\n\n 股票的平均收盘价: %.2f 元\n" ,aprice);
}

```

运行结果如图 2-17 所示。

```

请输入23个该股票的收盘价
13.46  13.49  13.57  13.45  13.53  14.01  14.32  14.39
14.16  14.11  14.35  14.42  14.50  14.57  14.63  14.36
14.29  14.68  14.67  14.28  13.58  13.62  13.63

由高到低排序的收盘价如下:
14.68  14.67  14.63  14.57  14.50  14.42  14.39  14.36
14.35  14.32  14.29  14.28  14.16  14.11  14.01  13.63
13.62  13.58  13.57  13.53  13.49  13.46  13.45

股票的平均收盘价: 14.09元

```

图 2-17 例 2.41 程序运行结果

例 2.42 编程将二维数组作为形式参数：在自定义函数中求一个月 n 个交易日内某只股票的平均收盘价，并用起泡法将每天的收盘价由高到低排序。主函数从键盘输入每天的日期及收盘价存放到二维数组中，并输出平均收盘价及该数组由大到小排序后的值。

日期 3.09 表示 3 月 9 日。

```

#include <stdio.h>
#define N 23 /*一个月内交易的天数*/
float sort(float price[ ][2], int days)
{int i, j;
 float t , aprice=0;
 for(i=0 ; i<days ; i++)
     aprice=aprice+price[i][1];
 aprice=aprice/days;          /*求平均收盘价*/
 /*以下双重循环起泡法实现由高到低排序*/
 for(j=0 ; j<days-1 ; j++)
     for(i=0 ; i<days-1-j ; i++)
         if (price[i][1]<price[i+1][1])

```

```
        {t=price[i][1];                                /*存放收盘价的两个元素进行交换*/
          price[i][1]=price[i+1][1];
          price[i+1][1]=t;
          t=price[i][0];                                /*存放日期的两个元素也进行交换*/
          price[i][0]=price[i+1][0];
          price[i+1][0]=t;
        }
    return (aprice);
}

main()
{int i ,j;
  float price[30][2], aprice;
  printf("请输入%d 个该股票的日期及收盘价\n" ,N);
  for (i=0; i<N; i++)
    for(j=0; j<2; j++)
      scanf("%f", &price[i][j]);
  aprice=sort(price, N);
  printf("\n\n 由高到低排序的收盘价如下: \n");
  printf("%-10s%-10s\n" , "日期" , "收盘价");
  for(i=0; i<N; i++)
    {for(j=0; j<2; j++)
      printf("%-10.2f",price[i][j]);      /*输出已排序的 N 个数组元素*/
      printf("\n") ;
    }
  printf("\n\n 股票的平均收盘价: %.2f 元\n" ,aprice);
}
```

运行结果如图 2-18 所示。

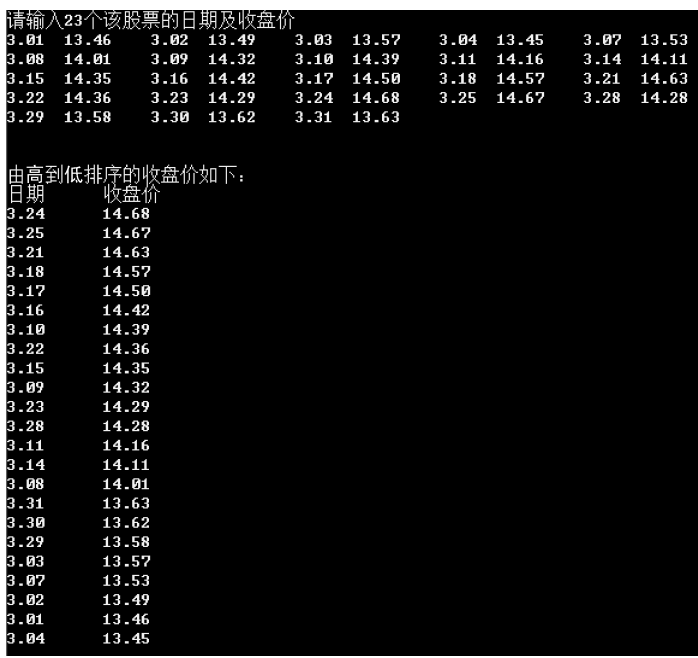


图 2-18 例 2.42 程序运行结果

例 2.43 编程实现函数的嵌套调用：从键盘上输入整数 n ，求 $1! + 2! + 3! + \cdots + n!$ 阶乘之和。

主函数及自定义函数分别完成以下不同的子任务。

主函数 main 调用 sub 求和函数，并输出阶乘之和。

函数 sub 调用 fac 函数求 $1! + 2! + 3! + \cdots + n!$ 。

函数 fac 求 $m!$ 。

程序如下：

```
#include <stdio.h>
/*函数 fac 求 m 的阶乘*/
double fac(int m)
{ int i;
  double t=1;
  for(i=1; i<=m; i++)
    t=t*i;
  return(t);
}

/*函数 sub 求 1~n 的阶乘之和*/
double sub(int n)
{ int i;
  double sum=0;
  for (i=1; i<=n; i++)
    sum=sum+fac(i);
  return(sum);
}

main()
{int n;
  double s;
  printf("请输入一个整数");
  scanf("%d", &n);
  s=sub(n);
  printf("1—%d 的阶乘之和: %.0f\n", n, s);
}
```

运行结果：

```
请输入一个整数 10
1—10 的阶乘之和: 4037913
```

以上程序总是把被调函数写在主调函数之前，所以可以省略被调函数的声明。

下列程序与例 2.43 在 3 个函数的书写顺序上正好相反，那么在主调函数中必须添加被调函数的声明。

```
#include <stdio.h>
main()
{int n;
```

```
double s;
double sub(int n); /*函数声明*/
printf("请输入一个整数");
scanf("%d", &n);
s=sub(n);
printf("1—%d 的阶乘之和: %.0f\n", n, s);
}
```

```
/*函数 sub 求 1~n 的阶乘之和*/
double sub(int n)
{ int i;
  double sum=0;
  double fac(int m); /*函数声明*/
  for (i=1; i<=n; i++)
    sum=sum+fac(i);
  return(sum);
}
```

```
/*函数 fac 求 m 的阶乘*/
double fac(int m)
{ int i;
  double t=1;
  for(i=1; i<=m; i++)
    t=t*i;
  return(t);
}
```

例 2.44 编程实现函数的递归调用和嵌套调用：从键盘上输入整数 n ，求 $1! + 2! + 3! + \dots + n!$ 阶乘之和。

主函数及自定义函数分别完成以下不同的子任务。

主函数 main 调用 sub 求和函数，并输出阶乘之和。

函数 sub 调用 factorial 函数求 $1! + 2! + 3! + \dots + n!$ 。

函数 factorial 递归函数求 $m!$ 。

程序如下：

```
#include <stdio.h>
double factorial(int m)
{ /*递归函数 factorial 求 m 的阶乘*/
  double t=1;
  if (m<=1)
    t=1;
  else
    t=m*factorial(m-1);
  return (t);
}

double sub(int n)
{ /*函数 sub 求 1~n 的阶乘之和*/
```

```

int i;
double sum=0;
for (i=1; i<=n; i++)
    sum=sum+factorial(i);
return(sum);
}

main()
{int n;
 double s;
 printf("请输入一个整数");
 scanf("%d", &n);
 s=sub(n);
 printf("1—%d 的阶乘之和: %.0f\n", n, s); /*%.0f 用来输出阶乘没有小数部分*/
}

```

运行结果:

```

请输入一个整数 10
1—10 的阶乘之和: 4037913

```

例 2.45 编程利用静态变量分别求 1~10 的阶乘。

程序如下:

```

#include <stdio.h>
double fact(int n)
{static double t=1; /*静态局部变量 t 中保留了上次调用结束时的值*/
 t=t*n;
 return (t);
}

main()
{int i;
 for(i=1; i<=10; i++)
    printf("%d!=%.0f\n",i,fact(i)); /*调用 fact 函数求分别求 1~10 的阶乘, 并输出*/
}

```

运行结果如图 2-19 所示。

思考: 如果把以上程序中 `static double t=1;` 改为 `double t=1;` 就会得到如图 2-20 所示的错误运行结果, 为什么?

```

1!=1
2!=2
3!=6
4!=24
5!=120
6!=720
7!=5040
8!=40320
9!=362880
10!=3628800

```

图 2-19 例 2.45 程序运行结果

```

1!=1
2!=2
3!=3
4!=4
5!=5
6!=6
7!=7
8!=8
9!=9
10!=10

```

图 2-20 修改例 2.45 程序后的运行结果

例 2.46 编程利用全局变量分别求 1~10 的阶乘。

程序如下：

```
#include <stdio.h>
double t=1;          /*全局变量 t*/
double facton(int n)
{t=t*n;              /*全局变量 t 中保留了上次调用结束时的值*/
  return (t);
}

main()
{int i;
  for(i=1; i<=10; i++)
    printf("%d!=%.0f\n",i,faction(i));/*调用 facton 函数分别求 1~10 的阶乘,并输出*/
}
```

运行结果如图 2-19 所示。

例 2.47 编程实现带参数的宏定义。

程序如下：

```
#include <stdio.h>
#define PT 5.5
#define s(x) PT*x*x
main()
{int a=1, b=2;
  printf("%4.1f \n", s(a+b));
}
```

运行结果：

9.5

以上程序是用实参 a+b 代替 PT*x*x 中的形参 x，成为 5.5*1+2*1+2。

2.8.3 实验项目

1. 实验目的和要求

- (1) 掌握函数定义和调用的方法。
- (2) 掌握数组名作函数参数的方法及意义。
- (3) 了解局部变量、全局变量的作用域和生存期。
- (4) 了解编译预处理命令的用法及意义。

2. 实验内容和步骤

(1) 编程利用主函数调用自定义函数，输出如图 2-21 所示的图形，要求屏幕背景为绿色，字符颜色为白色。可参考本章范例程序例 2.39。

提示：颜色设置库函数为 `system("color 27");`。



图 2-21 程序运行输出的图形

思考：若改背景为浅蓝色，字符颜色为浅黄色，并且从第 3 个制表位开始输出图形，程序应如何修改？

(2) 某大学教师工作量计算公式：工作量=系数×授课学时数，系数与教师的授课班级数有关，具体列表如下：

授课班级数	系数
1	1
2	1.5
3	1.9
4	2.3

编程利用主函数调用自定义函数，从键盘输入一名教师的授课学时数及班级数，求该教师的工作量。当从键盘输入的授课学时数为 0 时程序结束。可参考本章范例程序例 2.40。

(3) 编程将一维数组名作为形式参数：自定义函数统计 n 个学生中大于高考平均分的人数，并用起泡法将这些学生的高分由高到低排序。

主函数从键盘输入 30 个学生的高分存放在一维数组中，并显示大于高考平均分的人数及该数组由大到小排序后的值。可参考本章范例程序例 2.41。

(4) 编程将二维数组名作为形式参数：自定义函数统计 n 个学生中大于高考平均分的人数，并用起泡法将这些学生的高分由高到低排序。

主函数从键盘输入 30 个学生的学号及高分存放在二维数组中，并显示大于高考平均分的人数及该数组由大到小排序后的值。可参考本章范例程序例 2.42。

提示：学号只输入最后 4 位数字。

思考：要求从键盘输入 45 个学生的学号及高分存放在二维数组中，程序应如何修改？

(5) 编程实现函数的嵌套调用：从键盘上输入整数 n 。

当整数 n 是素数时，求 $1+2+3+\cdots+n$ 自然数之和。

当整数 n 不是素数时，求 $1!+2!+3!+\cdots+n!$ 阶乘之和。

可参考本章范例程序例 2.43 和范例程序例 2.22。

(6) 编程将 1~10 的阶乘的运算结果存放在一维数组中，并输出一维数组中是 10 的倍数的那些元素的值。

要求：自定义函数利用静态变量求 n 的阶乘。可参考本章范例程序例 2.45。

3. 实验思考和总结

(1) 解答范例程序或实验项目中的思考题。

(2) 总结编程调试的经验教训。

2.9 指针应用

2.9.1 同步自测

1. 若有定义：int x, *pb; 则下列正确的赋值表达式是 ()。

- A. pb=&x
- B. pb=x
- C. *pb=&x
- D. *pb=*x

2. 若有下列定义, 则 $p+7$ 表示 ()。

```
int a[10], *p=a;
```

- A. 元素 $a[6]$ 的地址 B. 元素 $a[6]$ 的值
C. 元素 $a[7]$ 的地址 D. 元素 $a[7]$ 的值

3. 若有定义: $\text{int } a[5][3]$, 则对数组 a 的第 i 行 j 列元素地址的正确引用是 ()。

- A. $*(a[i]+j)$ B. $(a+i)$ C. $*(a+j)$ D. $a[i+j]$

4. 下列选项中两个语句都正确的是 ()。

- A. $\text{char } s[8]; s=\{\text{"Beijing"}\};$ B. $\text{char } *s; s=\{\text{"Beijing"}\};$
C. $\text{char } s[8]; s=\text{"Beijing"};$ D. $\text{char } *s; s=\text{"Beijing"};$

5. 已定义下列函数

```
fun(char *p2, char *p1)
{ while ((*p2=*p1)!='\0') { p1++; p2++; } }
```

函数的功能是: ()。

- A. 将 $p1$ 所指向的字符串复制到 $p2$ 所指向的内存空间
B. 将 $p1$ 所指向的字符串的地址赋给指针 $p2$
C. 将 $p1$ 和 $p2$ 两个指针所指向的字符串进行比较
D. 检查 $p1$ 和 $p2$ 两个指针所指向的字符串中是否有 $\backslash 0$

6. 下列程序的输出结果是 ()。

```
#include <stdio.h>
main()
{int a[ ]={1,2,3,4,5,6}, *k[3], i=0;
 while(i<3)
 {k[i]=&a[2*i+1];
 printf("%d", *k[i]);
 i++; }
}
```

- A. 135 B. 123 C. 246 D. 456

7. 下列程序的输出结果是 ()。

```
#include <stdio.h>
main()
{int a[5]={10,20,30,40,50}, *p;
 p=&a[1];
 printf("%d", *p++); }
```

- A. 20 B. 30 C. 21 D. 31

8. 设有以下函数

```
void fun(int n, float s) {...}
```

则下面对函数指针的定义和赋值均是正确的是 ()。

- A. $\text{void } *pf(\text{float}, \text{int}); pf=\text{fun};$

- B. void (*pf)(int, float); pf=fun;
C. void *pf(int, float); *pf=fun;
D. void (*pf)(int, float); pf=&fun;

9. 下列程序的输出结果是 ()。

```
#include <stdio.h>
void sub( int x, int y, int *z )
{ *z=y-x ; }
main()
{ int a, b, c ;
  sub(10, 5, &a ) ;
  sub(7,a, &b ) ;
  sub( a, b, &c ) ;
  printf("%d,%d,%d\n", a, b, c ) ;
}
```

- A. 10,5,7 B. -10,-12,-5 C. -5,-12,-7 D. -5,-10,7

10. 下列程序的输出结果是 ()。

```
#include <stdio.h>
#include <string.h>
main()
{ char *s1="AbDeG";
  char *s2="AbdEg";
  s1+=2;
  s2+=2;
  printf("%d\n", strcmp(s1,s2));
}
```

- A. 正数 B. 负数 C. 0 D. 程序有错

11. 填空题：下列程序的输出结果是 ()。

```
#include <stdio.h>
main()
{ int x[8]={8,7,6,5,0,0}, *s;
  s=x+3;
  printf("%d\n",s[2]);
}
```

12. 填空题：下列程序的输出结果是 ()。

```
#include <stdio.h>
void ast( int x , int y , int *cp , int *dp )
{ *cp = x+y ;
  *dp = x-y ; }
main()
{ int a=4 , b=3 , c , d ;
  ast( a , b , &c , &d ) ;
  printf("c=%d,d=%d\n" , c , d ) ;
}
```

13. 填空题：下列程序的输出结果是（ ）。

```
#include <stdio.h>
main()
{char *s="APPLE";
  do
  {printf("%d",*s%10);s++;
  } while(*s);
}
```

14. 填空题：下列程序的输出结果是（ ）。

```
#include <stdio.h>
main()
{ char *a[ ]={"apple","pear"};
  int i;
  for(i=1;i<5;i++)
    printf("%c",*(a+i));
}
```

15. 填空题：下列程序的输出结果是（ ）。

```
#include <stdio.h>
main()
{char a[ ]="computer" , b[ ]="commerce";
  char *p1,*p2;
  int k;
  p1=a; p2=b;
  for(k=0; k<8; k++)
    if(*(p1+k)==*(p2+k)) printf("%c",*(p1+k));
}
```

16. 程序填空题：求数组元素的最小值及下标。

```
#include <stdio.h>
main()
{int a[10],*p,*s;
  for(p=a; p-a<10; p++)
    scanf("%d", p);
  for(p=a,s=a; p-a<【1】; p++) if(*p<*s) 【2】
    printf("最小值: %d 下标: %d\n" , *s,s-a);
}
```

17. 程序填空题：将数组元素逆序存放。

```
#include <stdio.h>
void invert(int *x, int n)
{int *p, temp, *i, *j, m;
  m=(n-1)/2;
  j=x+n-1;
  p=x+m;
```



```

    for (i=x; i <= 【1】; i++, j--)
    {temp=*i;
      【2】
      *j=temp;
    }
}
main()
{int i, a[10]={1,3,5,7,9,2,4,6,8,10};
  【3】
  for (i=0; i<10; i++)
    printf("%d  ", a[i]);
  printf("\n");
}

```

18. 判断题：在 C 语言中允许将一个整数赋给指针变量。()
19. 判断题：有一维数组 a[10]，那么 a 是 a[0]的地址。()
20. 判断题：char (*p)[10];定义了有 10 个元素的指针数组。()

2.9.2 范例程序解析

例 2.48 用指针变量作函数形参，判断一个正整数是否是完全平方数。若是则输出形式如下：81 是完全平方数 $81=9\times 9$ 。

程序如下：

```

#include <stdio.h>
#include <math.h>
void square(int *p1 , int *p2)
{int k;
  k=sqrt(*p1);          /*只有平方根的整数部分存入 k*/
  if(*p1==k*k)
  { *p2=1;              /*是完全平方数 f=1*/
    *p1=k;              /*将 n 的平方根赋值给 n*/
  }
}
main( )
{int n, n1, f=0;
  printf("请输入一个正整数");
  scanf("%d" , &n);
  n1=n;
  square(&n , &f);      /*n 的地址传递给 p1   f 的地址传递给 p2 */
  if (f==1)
    printf("%d 是完全平方数  %d=%d\times %d\n" , n1,n1,n,n);
  else
    printf("%d 不是完全平方数\n",n1);
}

```

思考：(1) 调用函数 square 之后，为什么 n 和 n1 的值不相等了？

(2) 为什么 k 定义为整型变量？

第 1 次运行结果：

请输入一个正整数 225
225 是完全平方数 225=15×15

第 2 次运行结果：

请输入一个正整数 79
79 不是完全平方数

例 2.49 主函数中两个数组分别指向或存放了下列 10 种食材的名称和每 100 克食材中微量元素锌的含量。

食材	锌含量（毫克）
驴肉	4.26
牡蛎	9.39
南瓜子	7.12
牛肉	7.61
山核桃	12.59
松子	9.02
西瓜子	7.12
章鱼	5.18
榛子	5.83
猪肉	2.99

要求主函数调用自定义函数按锌的含量由高到低用起泡法排序。指针变量、指针数组作形参。

程序如下：

```
#include <stdio.h>
#define N 10 /*食材种类数*/
void sort(float *p1, char *p2[ ], int n)
{int i,j;
 float t;
 char *food;
 /*下列双重循环用起泡法由高到低排序*/
 for(j=0; j<n-1; j++)
   for(i=0 ;i<n-1-j ;i++)
     if (*(p1+i)<*(p1+i+1))
       {t=*(p1+i); *(p1+i)=*(p1+i+1); *(p1+i+1)=t;
        food=p2[i]; p2[i]=p2[i+1]; p2[i+1]=food;
       }
}

main()
{int i;
 float a[N]={4.26,9.39,7.12,7.61,12.59,9.02,7.12,5.18,5.83,2.99};
 char *f[N]={"驴肉","牡蛎","南瓜子","牛肉","山核桃","松子","西瓜子","章鱼",
```

```

        "榛子","猪肉"};
sort(a, f, N); /*调用排序函数*/
printf("每 100 克食材含锌量（毫克）由高到低排序如下：\n");
for(i=0; i<N; i++)
    printf("%-16s%-8.2f\n",f[i],a[i]);
}

```

运行结果如图 2-22 所示。

食材名称	含锌量 (毫克)
山核桃	12.59
牡蛎	9.39
松子	9.02
牛肉	7.61
南瓜子	7.12
西瓜子	7.12
榛子	5.83
章鱼	5.18
驴肉	4.26
猪肉	2.99

图 2-22 例 2.49 运行结果

以上程序中的自定义函数也可修改如下，同样也可得到图 2-22 所示的结果。

```

void sort(float *p1, char *p2[ ], int n)
{int i,j;
  float t;
  char *food;
  for(j=0; j<n-1; j++)
      for(i=0 ;i<n-1-j ;i++)
          if (p1[i]<p1[i+1])
              {t=p1[i];    p1[i]=p1[i+1];    p1[i+1]=t;
                food=p2[i];    p2[i]=p2[i+1];    p2[i+1]=food;
              }
}

```

例 2.50 用选择法实现例 2.49 的功能，并要求输出结果为白底浅蓝字。

程序如下：

```

#include <stdio.h>
#include <stdlib.h>
#define N 10 /*食材种数*/
void sort(float *p1, char *p2[ ], int n)
{int i,j,k;
  float t;
  char *food;
  system("cls"); /*清除屏幕显示信息*/
  system("color F9"); /*设置屏幕背景为白色 字符颜色为蓝色*/
  /*下列双重循环用选择法由高到低排序*/
  for(i=0; i<n-1; i++)
  {k=i;
    for(j=i+1; j<n ;j++)

```

```
        if (*(p1+j)>*(p1+k))
            k=j;
        t=*(p1+k);    *(p1+k)=*(p1+i);    *(p1+i)=t;
        food=p2[k];    p2[k]=p2[i];        p2[i]=food;
    }
}

main()
{int i;
 float a[N]={4.26,9.39,7.12,7.61,12.59,9.02,7.12,5.18,5.83,2.99};
 char *f[N]={"驴肉","牡蛎","南瓜子","牛肉","山核桃","松子","西瓜子","章鱼",
             "榛子","猪肉"};
 sort(a, f, N); /*调用排序函数*/
 printf("每 100 克食材含锌量（毫克）由高到低排序如下：\n");
 for(i=0; i<N; i++)
     printf("%-16s%-8.2f\n",f[i],a[i]);
}
```

运行结果如图 2-23 所示。

每100克食材含锌量（毫克）由高到低排序如下：	
山核桃	12.59
牡蛎	9.39
松子	9.02
牛肉	7.61
南瓜子	7.12
西瓜子	7.12
榛子	5.83
章鱼	5.18
驴肉	4.26
猪肉	2.99

图 2-23 例 2.50 运行结果

例 2.51 二维数组中存放了 5 名学生 3 次高考模拟考试成绩，使用指向元素的指针变量引用数组元素，求最高分和平均分，并输出全部数组元素。

程序如下：

```
#include <stdio.h>
#define M 5 /*学生人数*/
#define N 3 /*每位学生有 3 个分数*/
main()
{int a[M][N]={{605,612,623},{541,601,598},{595,587,592},{639,602,641},
{646,657,648}};
 int max,*p;
 float aver;
 aver=0;
 max=0;
 printf("%d 名学生的%d 次模拟考试分数如下：",M,N);
 for (p=a[0]; p<a[0]+M*N; p++)
 {aver=*p+aver; /*使用*p 依次引用 a[0][0],a[0][1] 等 15 个元素*/
 if (*p>max)
```

```

    max=*p;
    if((p-a[0])%3==0) printf("\n"); /*每行输出 3 个元素之后换行*/
    printf("%6d" , *p);
}
aver=aver/(M*N);          /*求平均分*/
printf("\n\n 最高分%d    平均分%.1f\n" , max,aver);
}

```

运行结果如图 2-24 所示。

```

5名学生的3次模拟考试分数如下:
605  612  623
541  601  598
595  587  592
639  602  641
646  657  648
最高分657  平均分612.5

```

图 2-24 例 2.51 运行结果

例 2.52 二维数组 `a` 中存放了 5 名学生 3 次高考模拟考试成绩,使用一维数组名 `a[i]` 引用二维数组元素,求最高分和平均分,并输出全部数组元素。

程序如下:

```

#include <stdio.h>
#define M 5 /*学生人数*/
#define N 3 /*每位学生有 3 个分数*/
main()
{int a[M][N]={ {605,612,623},{541,601,598},{595,587,592},{639,602,641},
               {646,657,648}};
  int i,j,max;
  float aver;
  aver=0;
  max=0;
  printf("%d 名学生的%d 次模拟考试分数如下: " , M,N);
  for (i=0; i<M; i++)
    for(j=0; j<N; j++)
      {aver=(a[i]+j)+aver; /*使用*(a[i]+j)依次引用 a[0][0],a[0][1]等 15 个元素*/
       if(*(a[i]+j)>max)
         max=*(a[i]+j);
       if((a[i]+j-a[0])%3==0) printf("\n"); /*每行输出 3 个元素之后换行*/
       printf("%6d" , *(a[i]+j));
      }
  aver=aver/(M*N); /*求平均分*/
  printf("\n\n 最高分%d    平均分%.1f\n" , max,aver);
}

```

运行结果如图 2-24 所示。

思考: 以上程序中若将 `*(a[i]+j)` 可以改写为 `*(a+i+j)`, 程序可以得到正确结果吗?

例 2.53 一维数组中存放了 10 种食材每 100 克的含锌量（毫克），设计一个返回指针值的函数求含锌量的最高值。

程序如下：

```
#include <stdio.h>
#define N 10      /*食材种数*/
float *func( float *p , int n)
{int i;
 float *pmax;
 pmax=p;          /*将 a[0] 的地址赋给 pmax */
 for(i=0; i<n; i++)
 {if(*p>*pmax) pmax=p;
  p++;
 }
 return(pmax);    /*返回值是最高值的地址*/
}

main()
{float a[N]={4.26,9.39,7.12,7.61,12.59,9.02,7.12,5.18,5.83,2.99};
 float *pmax;
 pmax=func(a, N);
 printf("每百克含锌量最高值%.2f 毫克\n" , *pmax);
}
```

运行结果：

含锌量最高值 12.59 毫克

例 2.54 利用指针数组编程：将某教材 unit1 的单词以赋初值的形式存入二维数组中，用起泡法按字母 a~z 的顺序排序，并以白底蓝字输出。

程序如下：

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define M 42 /*单词个数*/
main()
{int i,j;
 char *t;
 char *p[ ]={{ "honest      adj 诚实的,正直的"}, {"brave      adj 勇敢的"}, \
 {"loyal      adj 忠诚的,忠心的"}, {"wise      adj 英明的,明智的,聪明的"}, \
 {"handsome  adj 英俊的,大方的,美观的"}, {"smart      adj 聪明的,漂亮的,敏捷的"}, \
 {"argue      vt 争论,辩论"}, {"solution  n 解答,解决办法,解决方案"}, \
 {"classical  adj 古典的,古典文学的"}, {"fond      adj 爱的,多情的,喜欢的"}, \
 {"fond of    喜欢,爱好"}, {"match      n 火柴"}, \
 {"match      n 火柴"}, {"mirror      n 镜子"}, \
 {"fry      vt&vi 油煎,油炸"}, {"gun      n 炮,枪"}, \
 {"hammer    n 锤子,槌"}, {"sawn      &vt&vi 锯"}, \
 {"rope      n 绳,索,绳索"}, {"compass  n 罗盘,指南针"}, \
 {"movie     n 电影"}, {"cast      vt&vi 投掷,投射,抛"}, \
 {"survive    vt 幸免于,从……中生还 vi 幸存"}, {"deserted  adj 荒芜的,荒废的"}, \
 {"hunt      vt&vi&n 打猎,猎取,搜寻"}, {"hunt for    搜索,追寻,寻找"}, \
```

```

{"in order to 为了"}, {"share vt&vi 分享,共有,分配 n 共享,份额"}, \
{"sorrow n 悲哀,悲痛"}, {"care about 担心,关心"}, \
{"feeling n. 触觉,知觉,感觉,情绪"}, {"such as 例如"}, \
{"airplane n 飞机"}, {"parachute n 降落伞"}, \
{"lie n 谎话,谎言"}, {"speech n 演说,讲话,语音"}, \
{"adventure n&vt&vi 冒险,冒险经历"}, {"notebook n 笔记本,笔记本式电脑"}, \
{"e-pal n 网友"}, {"drop sb a line 给某人写信(通常指写短信)"}, \
{"formal adj 正式的,正规的"}, {"error n 错误,差错"};
system("cls"); /*清除屏幕显示信息*/
system("color F9"); /*设置屏幕背景为白色 字符颜色为浅蓝色*/
for(j=0; j<M-1; j++)
    for(i=0 ;i<M-1-j ;i++)
        if (strcmp(p[i],p[i+1])>0)
            {t=p[i]; p[i]=p[i+1]; p[i+1]=t;} /*双重循环起泡法由小到大排序*/
for(i=0; i<M; i++)
    puts(p[i]); /*输出p[i]所指向的那个单词字符串*/
}

```

运行结果如图 2-25 所示。

思考: (1) 将例 2.54 和例 2.38 比较, 数组元素 $p[i]$ 与 $c[i]$ 有何区别?

(2) 例 2.54 中为什么不能用库函数 `strcpy` 完成 $p[i]$ 和 $p[i+1]$ 值的交换?

```

adventure n&vt&vi 冒险,冒险经历
airplane n 飞机
argue vt 争论,辩论
brave adj 勇敢的
care about 担心,关心
cast vt&vi 投掷,投射,抛
classical adj 古典的,古典文学的
compass n 罗盘,指南针
deserted adj 荒芜的,荒废的
drop sb a line 给某人写信(通常指写短信)
e-pal n 网友
error n 错误,差错
feeling n. 触觉,知觉,感觉,情绪
fond adj 爱的,多情的,喜欢的
fond of 喜欢,爱好
formal adj 正式的,正规的
fry vt&vi 油煎,油炸
gun n 炮,枪
hammer n 锤子,槌
handsome adj 英俊的,大方的,美观的
honest adj 诚实的,正直的
hunt vt&vi&n 打猎,猎取,搜寻
hunt for 搜索,追寻,寻找
in order to 为了
lie n 谎话,谎言
loyal adj 忠诚的,忠心的
match n 火柴
match n 火柴
mirror n 镜子
movie n 电影
notebook n 笔记本,笔记本式电脑
parachute n 降落伞
rope n 绳,索,绳索
saw n &vt&vi 锯
share vt&vi 分享,共有,分配 n 共享,份额
smart adj 聪明的,漂亮的,敏捷的
solution n 解答,解决办法,解决方案
sorrow n 悲哀,悲痛
speech n 演说,讲话,语音
such as 例如
survive vt 幸免于,从……中生存 vi 幸存
wise adj 英明的,明智的,聪明的

```

图 2-25 例 2.54 运行结果

例 2.55 使用指向由 5 个元素组成的一维数组的指针变量编程：有 8 名学生，各学习 4 门课程。使用指向由 5 个元素组成的一维数组的指针变量，引用 8 行 5 列的二维数组，计算每个学生的总分存放在最后一列，并输出 8 名学生的成绩。

程序如下：

```
#include <stdio.h>
#define N 8 /*学生人数*/
void sum(int (*p)[5], int n)
{int i, j;
/*下列双重循环求每个学生的总分*/
for (i=0; i<n; i++)
    {*(*(p+i)+4)=0; /*等价于 a[i][4]=0*/
    for(j=0; j<4; j++)
        *(*(p+i)+4)=*(*(p+i)+4)+*(*(p+i)+j); /*求第 i 行的总分*/
    }
/*下列双重循环输出每个学生成绩*/
for (i=0; i<N; i++)
    {for(j=0; j<5; j++)
        printf("%5d" , *(*(p+i)+j) );
    printf("\n");
    }
}

main()
{int a[N][5]={ {78,69,75,64,0},{92,86,80,79,0},{88,65,76,84,0},{72,89,80,65,0},\ {79,68,77,94,0},{98,86,80,89,0},{68,65,76,74,0},{92,88,86,95,0}};
sum(a , N);
}
```

运行结果如图 2-26 所示。

思考：为什么程序中第 1 个双重循环的内循环条件是 `j<4`，而第 2 个双重循环的内循环条件是 `j<5`？

78	69	75	64	286
92	86	80	79	337
88	65	76	84	313
72	89	80	65	306
79	68	77	94	318
98	86	80	89	353
68	65	76	74	283
92	88	86	95	361

图 2-26 例 2.55 运行结果

例 2.56 使用指向函数的指针变量调用自定义函数。编程实现下列功能：

某商业银行一年以上定期存款年利率如下：

存款项目	年利率
1 年定期存款	1.50%
3 年定期存款	2.10%
5 年定期存款	2.75%

利用主函数调用自定义函数，求到期时的本息之和，并以白底蓝字输出。当从键盘输入的存款数为 0 时程序结束。

程序如下:

```
#include <stdio.h>
#include <stdlib.h>
float func(float deposit, int years)
/*利率 rate , 存款本金 deposit , 存款年数 years, 本金及利息总和 total*/
{float rate , total;
switch (years)
{case 1: rate=0.015; break;
case 3: rate=0.021; break;
case 5: rate=0.0275; break;
}
total=deposit*rate*years;
return (total); /*返回值为利息*/
}
main()
{float (*p)(float , int); /*定义指向函数的指针变量 p */
float total, deposit;
int years;
system("cls"); /*清除屏幕显示信息*/
system("color F9"); /*设置屏幕背景为白色 字符颜色为浅蓝色*/
while (1)
{printf("请输入存款数 年数 (存款数输入 0 结束程序)");
scanf("%f%d", &deposit, &years);
if (deposit==0) break;
if (years==1 || years==3 || years==5)
{p=func; /*使变量 p 指向函数 func*/
/*下一句中 (*p)(deposit, years) 用变量 p 调用函数 func */
total=deposit+(*p)(deposit, years);
printf("存款本金: %.2f 元 年数: %d 本息之和: %.2f 元\n\n", deposit, years, total); }
else
printf("输入的年数有错, 请重新输入\n");
}
}
```

运行结果如图 2-27 所示。

```
请输入存款数 年数 <存款数输入 0 结束程序>3560 3
存款本金:3560.00元 年数:3 本息之和: 3784.28元

请输入存款数 年数 <存款数输入 0 结束程序>20000 1
存款本金:20000.00元 年数:1 本息之和: 20300.00元

请输入存款数 年数 <存款数输入 0 结束程序>0 0
```

图 2-27 例 2.56 运行结果

例 2.57 使用带有形参的 main 函数, 在命令行输入存款数和存款年数。编程实现下列功能:

某商业银行一年以上定期存款年利率如下:

存款项目	年利率
1 年定期存款	1.50%
3 年定期存款	2.10%
5 年定期存款	2.75%

利用主函数调用自定义函数，求到期时的本息之和，并以白底蓝字输出。
程序如下：

```
#include <stdio.h>
#include <stdlib.h>
float func(float deposit, int years)
/*利率 rate ,存款本金 deposit , 存款年数 years,本金及利息总和 total*/
{float rate , total;
 switch (years)
 {case 1: rate=0.015; break;
 case 3: rate=0.021; break;
 case 5: rate=0.0275; break;
 }
 total=deposit*rate*years;
 return (total); /*返回值为利息*/
}
main(int argc , char *argv[])
{float total, deposit;
 int years;
 system("color F9"); /*设置屏幕背景为白色 字符颜色为浅蓝色*/
 deposit=atof(argv[1]); /*将命令行输入的存款数字字符串转化成浮点数*/
 years=atoi(argv[2]); /*将命令行输入的年数字字符串转化成整数*/
 if (years==1 || years==3 || years==5)
 {total=deposit+func(deposit, years);
 printf("存款本金:%.2f 元 年数:%d 本息之和: %.2f 元 \n\n",deposit,years,
 total); }
 else
 printf("输入的年数有错, 请重新执行程序\n");
}
```

程序生成的可执行文件 p2t56.exe 存放于文件夹 D:\prg1032\debug 中。
运行结果如图 2-28 所示。

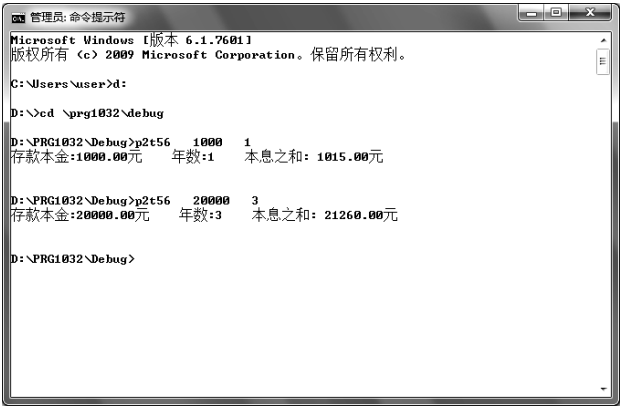


图 2-28 例 2.57 运行结果

2.9.3 实验项目

1. 实验目的和要求

- (1) 掌握指针的定义、指针变量的使用。
- (2) 掌握用指针作函数参数的编程方法。
- (3) 掌握指针与数组的关系，通过指针访问数组。
- (4) 掌握返回指针值的函数定义及调用方法。
- (5) 掌握使用指针数组来处理字符串数组的方法。
- (6) 了解使用指向函数的指针的使用方法。
- (7) 了解带有形参的 main 函数的编程及运行方法。

2. 实验内容和步骤

(1) 自定义函数：用指针变量作函数形参，判断一个正整数是否是完全平方数。

主函数：调用自定义函数找出 1~100 之间的完全平方数，并且输出形式如下：

81 是完全平方数 $81=9\times 9$ 可参考本章范例程序例 2.48。

(2) 主函数中两个数组分别指向或存放了下列 8 种食材的名称和每 100 克食材中维生素 C 的含量。

食材	维生素 C 含量（毫克）
猕猴桃	62
菠菜	32
菜花	61
橙子	47
苦瓜	56
芒果	23
木瓜	43
苋菜	47

要求主函数调用自定义函数按维生素 C 含量由高到低用起泡法或选择法排序，并求维生素 C 含量的平均值。指针变量、指针数组作形参。可参考本章范例程序例 2.49、例 2.50。

(3) 二维数组中存放了 10 名学生 C 语言课程的各项百分制成绩，其中第 0 列开始依次是实验成绩、作业成绩、试卷成绩。使用指向元素的指针变量引用数组元素，求第 3 列的综合成绩，并输出全部数组元素。

$$\text{综合成绩} = 0.15 \times \text{实验成绩} + 0.15 \times \text{作业成绩} + 0.7 \times \text{试卷成绩}$$

可参考本章范例程序例 2.51。

(4) 一维数组中存放了 8 家不同的商业银行 1 年期存款利率，设计一个返回指针值的函数求存款利率的最高值。可参考本章范例程序例 2.53。

(5) 利用指针数组编程：某学校选出 8 名优秀教师发言，介绍教书育人的先进事迹。要求将他们的姓名（汉字）按拼音 a~z 的顺序排列，以此作为先后发言的顺序。可参考本章范例程序例 2.54。

(6) 使用指向由 5 个元素组成的一维数组的指针变量编程：有 8 名学生，各学习 4 门课程。使用指向由 5 个元素组成的一维数组的指针变量，引用 8 行 5 列的二维数组，计算每个学生的平均分存放到最后一列，并输出 8 名学生的成绩。每个学生的 4 门课成绩输出不保留小数，平均分保留 1 位小数。

可参考本章范例程序例 2.55。注意定义 float 类型的二维数组。

(7) 使用指向函数的指针变量调用自定义函数。编程实现下列功能：某大学教师工作量计算公式：工作量=系数×授课学时数，系数与教师的授课班级数有关，具体列表如下：

授课班级数	系数
1	1
2	1.5
3	1.9
4	2.3

编程利用主函数调用自定义函数，从键盘输入一名教师的授课学时数及班级数，求该教师的工作量。当从键盘输入的授课学时数为 0 时程序结束。可参考本章范例程序例 2.56。

(8) 输入例 2.57 的源程序，编译、链接，然后在命令提示符窗口执行已生成的可执行文件。提示：在 Windows7 中，单击“开始”菜单→“附件”→“命令提示符”进入命令提示符窗口。

3. 实验思考和总结

- (1) 解答范例程序或实验项目中的思考题。
- (2) 总结编程调试的经验教训。

2.10 结构体应用

2.10.1 同步自测

- 1. C 语言规定结构体变量系统分配给它的存储空间是（ ）。
 - A. 该结构体中最后一个成员所需的存储空间
 - B. 该结构体中所有成员所需存储空间的总和
 - C. 该结构体中第一个成员所需的存储空间
 - D. 该结构体中占用最大存储空间的成员所需的存储空间

2. 有以下说明语句：

```
struct stu
{ int a;
float b;
} stutype;
```

则下面的叙述不正确的是（ ）。

- A. struct 是结构体类型的关键字
- B. struct stu 是用户定义的结构体类型名
- C. stutype 是用户定义的结构体类型名
- D. a 和 b 都是结构体成员名

3. 有以下说明语句, 则结构变量 s 的成员 num 的不正确引用是 ()。

```
struct student
{ int num;
  int age;
} s , *p;
```

4. 以下各选项说明一种新的类型名, 其中正确的是 ()。

- A. typedef integer int; B. typedef integer=int;
C. typedef int integer; D. typedef int=integer;

5. 在 VC++ 6.0 中使用 C 语言时, 若有如下定义:

```
struct data
{ long i;
  char ch;
  double f;
} k;
```

- 则结构变量 k 占用内存的字节数是 ()。

- A. 4 B. 8 C. 12 D. sizeof(struct data)

6. 在 VC++ 6.0 中, 下列程序的输出结果是 ()。

- A. 10 B. 结构体变量所占字节数
C. 程序有错, 无输出结果 D. 16

```
typedef struct
{ long x[2];
  int y[4];
  char z[8];
} Newtype;
Newtype data;
main()
{ printf("%d\n", sizeof(data)); }
```

7. 有以下说明和语句, 则值为 6 的表达式是 ()。

- A. p++->n B. p->n++ C. (*p).n++ D. ++p->n

```
Struct k
{ int n;
  struct k *next; };
struct k, a[3] , *p;
a[0].n=5; a[0].next=&a[1];
a[1].n=7; a[1].next=&a[2];
a[2].n=9; a[0].next='\0';
p=&a[0];
```

8. 下列程序的输出结果是 ()。

```
#include <stdio.h>
#include <string.h>
```

```

struct A
{ int a; char b[10]; double c;};
struct A f(struct A t);
main()
{struct A a={1111,"haier",1098.0};
a=f(a);
printf("%d,%s,%6.1f\n",a.a,a.b,a.c);
}
struct A f(struct A t)
{t.a=1002;
strcpy(t.b,"xiaoya");
t.c=1202.0;
return t; }

```

A. 1111,haier,1098.0

B. 1111,haier,1202.0

C. 1002,xiaoya,1202.0

D. 1002,xiaoya,1098.0

9. 下列程序的输出结果是 ()。

```

#include <stdio.h>
typedef struct
{int num;
double s;} LX;
void func( LX x )
{ x.num=78; x.s=86.2; }
main()
{LX a={66, 81.3};
func(a);
printf("%d\n",a.num);
}

```

A. 81.3

B. 66

C. 78

D. 86.2

10. 下列程序的输出结果是 ()。

```

#include <stdio.h>
struct ord
{ int x, y; } dt[2]={1,2,3,4};
main()
{struct ord *p=dt;
printf("%d,",++p->x);
printf("%d\n",++p->y);
}

```

A. 1,2

B. 2,3

C. 3,4

D. 4,1

11. 下列程序的输出结果是 ()。

```

struct s {int n;
char *c; } *p;
char d[ ]={'a','b','c','d','e'};

```

```

struct s a[]={10,&d[0],20,&d[1],30,&d[2],40,&d[3],50,&d[4]};
#include <stdio.h>
main()
{p=a;
printf("%d\n",++p->n);
printf("%d\n", (++p)->n);
printf("%c\n",++(*p->c));
}

```

- | | | | |
|-------|-------|-------|-------|
| A. 11 | B. 11 | C. 10 | D. 10 |
| 20 | 20 | 20 | 20 |
| c | b | c | b |

12. 运行下列程序段, 输出结果是 ()。

```

struct country
{int num;
Char name[10];
} x[5]={1,"China",2,"USA",3,"France",4, "England",5, "Spanish"};
struct country *p;
p=x+3;
printf("%d,%c",p->num, (*p).name[2]);

```

- | | | | |
|--------|--------|--------|--------|
| A. 3,a | B. 4,g | C. 2,U | D. 5,S |
|--------|--------|--------|--------|

13. 下列定义, 能输出 Adam 的语句是 ()。

```

Struct person
{char name[9];
int age;
};
struct person class[5]={"John",17,"Paul",19,"Mary",18,"Adam",16};

```

- A. printf("%s\n",class[1].name);
 B. printf("%s\n",class[2].name);
 C. printf("%s\n",class[3].name);
 D. printf("%s\n",class[0].name);

14. 若有如下定义, 则下列选项中赋值语句错误的是 ()。

```

struct data
{ int real,unreal;} data1={88,99},data2;

```

- A. data2=data1;
 B. data2=(2,6);
 C. data2.real=data1.real;
 D. data2.real=data1.unreal;

15. 下列程序输出结果是 ()。

```

#include<stdio.h>
main()

```

```

{struct student
{int num ;
int age;
} ;
struct student stu[3]={1001,20} , {1002,19} , {1003,21}} ;
struct student *p ;
p=stu ;
printf("%d" , (++p).age);}

```

- A. 21 B. 1002 C. 19 D. 20

16. 填空题：下列程序的输出结果是（ ）。

```

#include <stdio.h>
struct tree
{int x;
char *s; } t;
void func(struct tree t)
{ t.x=10;
t.s="computer"; }
main()
{t.x=1;
t.s="minicomputer";
func(t);
printf("%d,%s\n",t.x,t.s);
}

```

17. 填空题：下列程序输出结果是（ ）。

```

#include<stdio.h>
main()
{struct s
{int a1 ;
struct s *a2 ;
} ;
struct s a[3]={1 , &a[1] , 2 , &a[2] , 3 ,&a[0] } , *ptr ;
ptr=&a[1] ;
printf("%d" , ++ptr->a1);}

```

18. 程序填空题：下列程序的功能是输出结构体变量 s 所占字节数及 s 的值。

```

#include <stdio.h>
main()
{struct stu
{long num;
char name[10];
} s={2015001, "张凯"};
printf("变量 s 占字节数: %d\n" , 【1】(struct stu));
printf("学号: %ld 姓名: %s \n" , 【2】);
}

```


19. 程序填空题：下列程序实现的是输出结构体数组中年龄居中者的姓名和年龄。

```
#include <stdio.h>
main()
{struct student
{char name[10];
int age;
};
struct student a[ ]={"王雪婷", 21, "宋慧敏", 18, "李雅静", 20};
int i,max,min;
max=min=a[0].age;
for(i=0; i<3; i++)
if(a[i].age>max) 【1】
else if (a[i].age<min) 【2】
for(i=0; i<3; i++)
if((a[i].age!=max) && (a[i].age!=min))
printf("年龄居中者的姓名 年龄: %s %d \n" , 【3】);
}
```

20. 程序填空题：对候选人得票的统计程序。设有 3 个候选人，每次输入一个得票的候选人名字，最后输出各人得票结果。

```
#include <stdio.h>
【1】
struct person
{char name[20];
int count;
} leader[3]={"李明",0,"张莉",0,"王伟",0};
main()
{int i, j;
char leader_name[20];
for(i=1; i<=10; i++)
{printf("请输入一个姓名");
scanf("%s", 【2】 );
for(j=0; j<3; j++)
if (strcmp(leader_name, leader[j].name) 【3】)
leader[j].count++;
}
printf("\n");
for(i=0; i<3; i++)
printf("%6s:%d\n", leader[i].name,leader[i].count);
}
```

2.10.2 范例程序解析

例 2.58 从键盘输入 5 名学生的姓名、性别以及英语、数学、语文 3 门课成绩，求每个人的平均分，并按平均分由高到低排序、输出。

程序如下:

```
#include <stdio.h>
#define N 5 /*学生人数*/
struct student
{char name[9];
char sex[3];
int score[3];
float avr;
} stu[N] , s;

main()
{int i, j, sum;
for(i=0; i<N; i++)
{printf("请输入第%d 个学生的姓名 性别\n",i+1);
scanf("%s%s", stu[i].name , stu[i].sex);
printf("请依次输入%s 3 门课的成绩\n", stu[i].name);
for(j=0; j<3; j++)
scanf("%d", &stu[i].score[j]);
}
/*以下双重循环求每个学生的平均分*/
for(i=0; i<N ; i++)
{sum=0;
for(j=0; j<3; j++)
sum=sum+stu[i].score[j];
stu[i].avr=sum/3.0; }
/*以下双重循环按平均分由高到低用起泡法排序*/
for(j=0; j<(N-1); j++)
for(i=0; i<(N-1-j); i++)
if (stu[i].avr < stu[i+1].avr)
{ s=stu[i]; stu[i]=stu[i+1]; stu[i+1]=s; }
/*以下程序按名次先后输出 N 个学生的信息*/
j=1; /*j 表示名次*/
printf("%-9s%-9s%-9s%-9s%-9s%-9s\n"
,"名次","姓名","性别","英语","数学","语文","平均分");
printf("%-9d%-9s%-9s", j , stu[0].name, stu[0].sex);
printf("%-9d%-9d%-9d%-9.1f\n",
stu[0].score[0], stu[0].score[1], stu[0].score[2],stu[0].avr);
for(i=1; i<N; i++)
{if (stu[i].avr!=stu[i-1].avr) j++; /*平均分相同的名次也相同*/
printf("%-9d%-9s%-9s", j, stu[i].name, stu[i].sex);
printf("%-9d%-9d%-9d%-9.1f\n",
stu[i].score[0], stu[i].score[1], stu[i].score[2],stu[i].avr);
}
}
```

运行结果如图 2-29 所示。

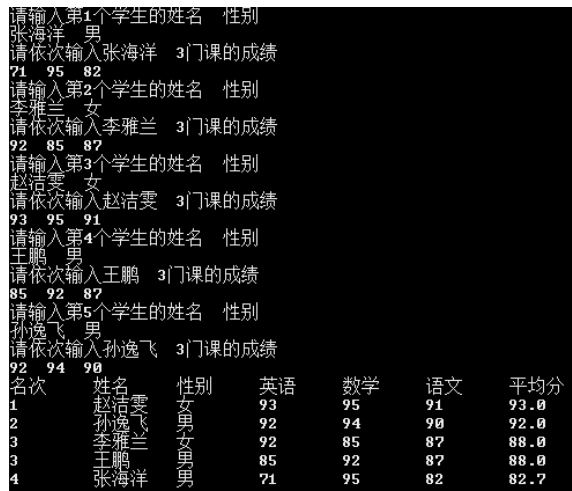


图 2-29 例 2.58 运行结果

例 2.59 某公司推销人员的基本信息包括：工号、姓名、底薪、补贴、销售额、应发工资。

应发工资=底薪+补贴+销售额×0.05

要求输入 5 名推销人员的工号、姓名、底薪、补贴、销售额，计算出应发工资且输出每位员工的上述基本信息。

程序如下：

```
#include <stdio.h>
#define N 5 /*人数*/
main()
{int i;
 struct person
 {char gh[6]; /*工号*/
 char xm[9]; /*姓名*/
 float dx; /*底薪*/
 float bt; /*补贴*/
 float xse; /*销售额*/
 float yfgz; } s[N]; /*应发工资*/
 for(i=0; i<N; i++)
 {printf("请依次输入第%d 的人的工号 姓名 底薪 补贴 销售额\n" , i+1);
 scanf("%s%s%f%f%f", s[i].gh,s[i].xm,&s[i].dx,&s[i].bt,&s[i].xse);
 s[i].yfgz=s[i].dx+s[i].bt+0.05*s[i].xse; /*计算应发工资*/
 }
 printf("%-9s%-9s%-9s%-9s%-9s%-9s\n", "工号","姓名","底薪","补贴","销售额","应发工资");
 for( i=0; i<N; i++)
 printf("%-9s%-9s%-9.2f%-9.2f%-9.2f%-9.2f\n" , s[i].gh,s[i].xm,s[i].dx,s[i].bt, s[i].xse, s[i].yfgz);
 }
```

运行结果如图 2-30 所示。

请依次输入第1的人的工号	姓名	底薪	补贴	销售额	
1001 张霞	2100	200	45000		
请依次输入第2的人的工号	姓名	底薪	补贴	销售额	
1002 李智睿	2100	200	67000		
请依次输入第3的人的工号	姓名	底薪	补贴	销售额	
1003 张小楠	3000	200	86500		
请依次输入第4的人的工号	姓名	底薪	补贴	销售额	
1004 刘彤彤	3000	230	81600		
请依次输入第5的人的工号	姓名	底薪	补贴	销售额	
1005 王玉玲	2000	200	43800		
工号	姓名	底薪	补贴	销售额	应发工资
1001	张霞	2100.00	200.00	45000.00	4550.00
1002	李智睿	2100.00	200.00	67000.00	5650.00
1003	张小楠	3000.00	200.00	86500.00	7525.00
1004	刘彤彤	3000.00	230.00	81600.00	7310.00
1005	王玉玲	2000.00	200.00	43800.00	4390.00

图 2-30 例 2.59 运行结果

例 2.60 某居民楼 2 单元 6 户居民的全年用电量基本信息如下：

户号、姓名、全年电量、第一挡电量、第二挡电量、第三挡电量、全年电费。

从键盘输入 6 户居民的全年电量、户号、姓名，分别计算第一挡电量、第二挡电量、第三挡电量、全年电费，并输出 6 户居民的基本信息。

居民阶梯电价分为三挡：

第一挡电量：每户年用电量≤2520 度，收费每度 0.55 元；

第二挡电量：2520 度<每户年用电量≤4800 度，收费每度 0.60 元；

第三挡电量：每户年用电量>4800 度，收费每度 0.85 元。

程序如下：

```
#include <stdio.h>
#define N 6
main()
{int i;
struct family
{char user[9];
char name[9];
float elec[4];
float money;
} a[N];
/*以下循环输入 N 户家庭的 户号 姓名 全年用电量 存入数组 a */
for(i=0; i<N; i++)
{printf("请依次输入第%d 户家庭的 户号 姓名 全年用电量 \n" , i+1);
scanf("%s%s%f" , a[i].user, a[i].name,&a[i].elec[0]);
}
printf("\n%-10s%-10s%-10s%-10s%-10s%-10s%-10s\n" , "户号" , "姓名", "\n\n"
"全年电量", "第一挡", "第二挡", "第三挡", "全年电费");
/*以下循环计算电费及各挡的电量，并输出*/
for(i=0; i<N; i++)
{ if (a[i].elec[0]<=2520)
{a[i].money=a[i].elec[0]*0.55; /*年用电量≤2520 度*/
a[i].elec[1]=a[i].elec[0];
a[i].elec[2]=a[i].elec[3]=0;
```

```
    }
else if (a[i].elec[0]<=4800)
{a[i].money=2520*0.55+(a[i].elec[0]-2520)*0.6; /*2520 度<年用电量≤4800 度*/
a[i].elec[1]=2520;
a[i].elec[2]=a[i].elec[0]-2520;
a[i].elec[3]=0;
}
else
{a[i].money=2520*0.55+(4800-2520)*0.6+(a[i].elec[0]-4800)*0.85;
/*年用电量>4800 度*/

a[i].elec[1]=2520;
a[i].elec[2]=4800-2520;
a[i].elec[3]=a[i].elec[0]-4800;
}
printf("%-10s%-10s" , a[i].user,a[i].name);
printf("%-10.2f%-10.2f%-10.2f%-10.2f%-10.2f\n" \
,a[i].elec[0],a[i].elec[1],a[i].elec[2],a[i].elec[3], a[i].money);
}
}
```

运行结果如图 2-31 所示。



图 2-31 例 2.60 运行结果

例 2.61 参加速录师考试的 10 名考生的考号、姓名、速度已存入结构体数组中，根据考生的录入速度求其职业等级。具体等级如下：

速度（每分钟输入汉字个数）	职业等级
速度<140	不合格
140≤速度<180	初级速录员
180≤速度<220	中级速录师
速度≥220	高级速录师

主函数调用自定义函数按速度由快到慢排序并输出。

主函数调用自定义函数输出排序后的结构体数组。

程序如下:

```
#include    <stdio.h>
#include    <string.h>
#define    N    10                /*学生人数*/
struct    student
{char    num[9];                /*考号*/
  char    name[9];                /*姓名*/
  int    speed;                /*速度*/
  char    level[11];                /*职业等级*/
} ;
main()
{void    sort(struct    student    s[ ], int    n);                /*函数声明*/
  void    print(struct    student    s[ ], int    n);                /*函数声明*/
  int    i;
  struct    student    *p;
  struct    student    s[N]={ "20151001", "于雅静", 152, " " , "20151002", "张玉洁",
                                138, " " , \
  "20151003", "李天宇", 193, " " , "20151004", "李明月", 226, " " , \
  "20151005", "王鹏飞", 231, " " , "20151006", "崔晓琳", 186, " " , \
  "20151007", "范志刚", 192, " " , "20151008", "张云飞", 220, " " , \
  "20151009", "赵婷", 205, " " , "20151010", "李浩宇", 218, " " };
  for(i=0; i<N; i++)
    if(s[i].speed<140)
      strcpy(s[i].level, "不合格");
    else if (s[i].speed<180)
      strcpy(s[i].level, "初级速录员");
    else if (s[i].speed<220)
      strcpy(s[i].level, "中级速录师");
    else strcpy(s[i].level, "高级速录师");
  p=s;                /*指针变量指向结构体数组 s 的起始地址*/
  sort(p, N);                /*调用排序函数*/
  print(p, N);                /*调用输出结构体数组 s 的函数*/
}

/*以下 sort 函数实现的是用起泡法按速度由快到慢排序*/
void sort(struct student s[ ], int n)
{int i, j;
  struct student temp ;
  for(j=0 ; j<n-1 ; j++)
    for(i=0 ; i<n-1-j ; i++)
      if (s[i].speed<s[i+1].speed)
        {temp.speed=s[i].speed; /*交换 s[i] 和 s[i+1] 的全部成员的值*/
          s[i].speed=s[i+1].speed;
          s[i+1].speed=temp.speed;
          strcpy(temp.num, s[i].num);
          strcpy(s[i].num , s[i+1].num);
          strcpy(s[i+1].num, temp.num);
        }
```

```
strcpy(temp.name,s[i].name);
strcpy(s[i].name , s[i+1].name);
strcpy(s[i+1].name,temp.name);
strcpy(temp.level,s[i].level);
strcpy(s[i].level , s[i+1].level);
strcpy(s[i+1].level,temp.level);
}
}

/*以下 print 函数的功能是输出结构体数组*/
void print(struct student s[ ] , int n)
{int i;
printf("%-12s%-12s%-12s%-12s\n" , "考号" , "姓名" , "速度" , "职业等级");
for(i=0; i<N; i++)
{printf("%-12s%-12s" ,s[i].num , s[i].name);
printf("%-12d%-12s\n" ,s[i].speed , s[i].level); }
}
```

运行结果如图 2-32 所示。

思考：定义 struct student 数据类型的语句能移到 main 函数内部吗？为什么？

考号	姓名	速度	职业等级
20151005	王鹏飞	231	高级速录师
20151004	李明月	226	高级速录师
20151008	张云飞	220	高级速录师
20151010	李浩宇	218	中级速录师
20151009	赵婷	205	中级速录师
20151003	李天宇	193	中级速录师
20151007	范志刚	192	中级速录师
20151006	崔晓琳	186	中级速录师
20151001	于雅静	152	初级速录
20151002	张玉洁	138	不合格

图 2-32 例 2.61 运行结果

2.10.3 实验项目

1. 实验目的和要求

- (1) 掌握结构体类型、结构体变量的定义和引用方法。
- (2) 掌握结构体类型数组作形参的函数的定义及调用方法。

2. 实验内容和步骤

(1) 从键盘输入 5 名学生的姓名、性别，以及英语、数学、语文 3 门课成绩，求每个人的总分，并按分数由高到低排序、输出。可参考本章范例程序例 2.58。

(2) 某公司推销人员的基本信息包括：工号、姓名、底薪、补贴、销售额、应发工资。

应发工资=底薪+补贴+销售额×0.05

要求输入 5 名推销人员的工号、姓名、底薪、补贴、销售额。
计算出应发工资，且输出每个人的上述基本信息。

计算出应发工资的平均数，及大于平均数的人数。可参考本章范例程序例 2.59。

(3) 某居民楼 3 单元 10 户居民的全年用水量基本信息如下：

户号、姓名、全年用水量、第一挡水量、第二挡水量、第三挡水量、全年水费。

从键盘输入 10 户居民的全年用水量、户号、姓名，分别计算第一挡用水量、第二挡用水量、第三挡用水量、全年水费，并输出 10 户居民的基本信息。

居民用户按年度用水量计算，全年用水量划分为三挡，水价逐挡递增，按挡计收：

第一阶梯用水量不超过 144（含）立方米，水价为每立方米 4.2 元；

第二阶梯用水量为 144～288（含）立方米，水价为每立方米 5.6 元；

第三阶梯用水量为 288 立方米以上，水价为每立方米 9.8 元。

可参考本章范例程序例 2.60。

(4) 某市参加中考立定跳远的 10 名考生的考号、姓名、性别、跳远长度已存入结构体数组中，根据考生的性别、跳远长度求其结构体数组中分数成员的值。具体评分标准如下：

男	女	分数
2.41 米以上	1.91 米以上	10 分
2.21～2.40 米	1.73～1.90 米	8 分
1.93～2.20 米	1.49～1.72 米	6 分
1.73～1.92 米	1.33～1.48 米	4 分
1.72 米以下	1.32 米以下	2 分

编程要求：

主函数调用自定义函数按跳远长度由大到小排序并输出。

主函数调用自定义函数输出排序后的结构体数组。

可参考本章范例程序例 2.61。

3. 实验思考和总结

(1) 解答范例程序或实验项目中的思考题。

(2) 总结编程调试的经验教训。

2.11 文件应用

2.11.1 同步自测

- 1. C 语言文件操作函数 fread(buffer,size,n,fp)的功能是（ ）。
 - A. 从文件 fp 中读 n 个字节存入 buffer 中
 - B. 从文件 fp 中读 n 个大小为 size 字节的数据项存入 buffer 中
 - C. 从文件 fp 中读入 n 个字节放入大小为 size 字节的缓冲区 buffer 中
 - D. 从文件 fp 中读入 n 个字符数据放入 buffer 中
- 2. 下列关于 C 语言数据文件的叙述中正确的是（ ）。
 - A. C 语言只能读/写文本文件
 - B. C 语言只能读/写二进制文件

- C. 文件由字符序列组成, 可按数据的存放形式分为二进制文件和文本文件
D. 文件由二进制数据序列组成
3. 若有程序段 `FILE *fp; fp=fopen("file1.txt", "r");`, 则以下说法中正确的是 ()。
- A. `fp` 指向磁盘文件 `file1.txt` 的地址
B. `fp` 指向磁盘文件 `file1.txt` 所对应的 `FILE` 结构
C. 如果 `fp` 不为 `NULL`, 表示文件打开失败
D. 可以利用 `fp` 指针对文件 `file1.txt` 进行写操作
4. 若要用 `fopen` 函数打开一个新的二进制文件, 该文件要求既能读也能写, 则打开方式字符串应是 ()。
- A. "ab+" B. "wb+" C. "rb+" D. "ab"
5. `fscanf` 函数的正确调用形式是 ()。
- A. `fscanf(fp, 格式字符串, 输出表列);`
B. `fscanf(格式字符串, 输出表列, fp);`
C. `fscanf(格式字符串, 文件指针, 输出表列)`
D. `fscanf(文件指针, 格式字符串, 输入表列)`
6. `fwrite` 函数的一般调用形式是 ()。
- A. `fwrite(butter, count, size fp);`
B. `fwrite(fp, size, count, butter);`
C. `fwrite(fp, count, size, butter);`
D. `fwrite(butter, size, count, fp);`
7. 设文件中以二进制形式存有许多学生的数据, 且已经正确打开, 文件指针定位在文件开头, 若要从文件中读出 10 个学生的数据存入 `stu` 数组中, 则以下正确的语句是 ()。
- A. `fread(stu, sizeof(struct std), 10, fp);`
B. `fread(&stu[i], sizeof(struct std), 1, fp);`
C. `fread(stu+i, sizeof(struct std), 1, fp);`
D. `fread(stu, struct std, 10, fp);`
8. 函数调用语句 `fseek(fp, -30L, 2);` 的含义是 ()。
- A. 将文件位置指针移动到距离文件头 30 个字节处
B. 将文件位置指针从当前位置向后移动 30 个字节
C. 将文件位置指针从文件末尾处后退 30 个字节
D. 将文件位置指针移动到离当前位置 30 个字节处
9. 利用 `fseek` 函数正确的是 ()。
- A. `fseek(文件类型指针, 起始点, 位移量);` B. `fseek(fp, 位移量, 起始点);`
C. `fseek(位移量, 起始点, fp);` D. `fseek(起始点, 位移量, 文件类型指针);`
10. 在执行 `fopen` 函数时, `ferror` 函数的初值是 ()。
- A. TURE B. -1 C. 1 D. 0
11. 若 `fp` 是指向某文件的指针, 且已读到文件末尾, 则库函数 `feof(fp)` 的返回值是 ()。
- A. EOF B. -1 C. 非零值 D. NULL

12. 设有以下结构体类型:

```
struct st
{char name[8];
 int num;
 float s[4];
} student [20];
```

并且结构体数组 student 中的元素都已经有了值,若要将这些元素写到 fp 所指向的磁盘文件中,以下错误的形式是 ()。

- A. fwrite (student , sizeof(struct st),20 , fp);
- B. fwrite (student ,20* sizeof(struct st),1, fp);
- C. fwrite (student , 10*sizeof(struct st),10 , fp);
- D. for (i=0;i<20;i++)
 fwrite (student[i] , sizeof(struct st),1 , fp);

13. 有下列程序,若文本文件 file1.txt 中原来内容为: apple,则运行以上程序后,文件中的内容为 ()。

- A. applecomputer
- B. computerapple
- C. computer
- D. apple

```
#include <stdio.h>
main()
{FILE *fp;
 fp=fopen("file1.txt","w");
 fprintf(fp, "computer");
 fclose(fp);
}
```

14. 填空题: 有下列程序,程序的运行结果是 ()。

```
#include <stdio.h>
main()
{FILE *fp;
 int a[10]={7,8,9},i,n;
 fp=fopen("dl.dat" , "w");
 for(i=0;i<3;i++)
    fprintf(fp,"%d",a[i]);
 fprintf(fp,"\n");
 fclose(fp);
 fp=fopen("dl.dat","r");
 fscanf(fp,"%d",&n);
 fclose(fp);
 printf("%d\n",n);
}
```

15. 填空题：有下列程序，程序的运行结果是（ ）。

```
#include <stdio.h>
main()
{FILE *fp;
int s[10]={5,6,7,2,1},i;
fp=fopen("filea.dat" , "wb");
fwrite(s,sizeof(int),5,fp);
fwrite(s,sizeof(int),5,fp);
fclose(fp);
fp=fopen("filea.dat","rb");
fread(s,sizeof(int),10,fp);
fclose(fp);
for(i=0; i<10; i++) printf("%d" , s[i]);
}
```

16. 填空题：下列程序运行结果是（ ）。

```
#include <stdio.h>
main()
{FILE *fp;
char s1[]="shandong", s2[]="hainan";
if((fp=fopen("test.dat","wb"))==NULL)
{printf("Can' t open test.txt file\n");
exit(1);}
fwrite(s1,8,1,fp);
fseek(fp,0L,SEEK_SET);
fwrite(s2,6,1,fp);
fclose(fp);
if((fp=fopen("test.dat","rb"))==NULL)
{printf("Can' t open test.txt file\n");
exit(1);}
fread(s1, 8,1,fp);
printf("%s\n" , s1);
}
```

17. 程序填空题：下列程序由键盘输入字符，存放到文件文本文件 file1.txt 中，用@结束输入。

```
#include <stdio.h>
#include <stdlib.h>
main()
{ FILE *fp;
char ch ;
if ((fp=fopen("file1.txt" , 【1】))==NULL)
{ printf ("不能打开文件\n");
exit(0);
}
printf("请输入字符");
while( 【2】 !='@')
fputc( 【3】 ) ;
fclose(fp);
}
```

18. 程序填空题：下列程序将文本文件 file1.txt 中字符逐个读取并统计文件中字符的个数。

```
#include <stdio.h>
#include <stdlib.h>
main()
{FILE *fp ;
 long count=0;
 if((fp=fopen(【1】 , "r"))==NULL)
 {printf("不能打开文件\n"); exit(0); }
 rewind(fp);
 while (!feof(fp))
 {fgetc(【2】);
 count++;}
 printf("count=%ld\n",count);
 【3】
}
```

19. 判断题：fopen("fname.txt","r")的作用是打开一个磁盘文件，此文件可以被执行写操作。()

20. 判断题：当文件操作完成后必须使用 fclose 函数关闭文件。()

2.11.2 范例程序解析

例 2.62 从键盘上输入 5 个学生的姓名、学号、年龄、住址存入磁盘文件中。

程序如下：

```
#include <stdio.h>
#define N 5
struct student
{char name[10];
 int num;
 int age;
 char addr[30];
};
void save(struct student s[ ] , int n )
{FILE *fp;
 int i;
 if ((fp=fopen("stu.dat","wb"))==NULL)
 {printf("不能打开文件\n");
 exit(0); }
 for(i=0; i<n; i++)
 if (fwrite(&s[i], sizeof(struct student) , 1 ,fp)!=1)
 printf("写文件错误\n");
 fclose(fp);
}

main()
{ struct student s[N] , *p;
```

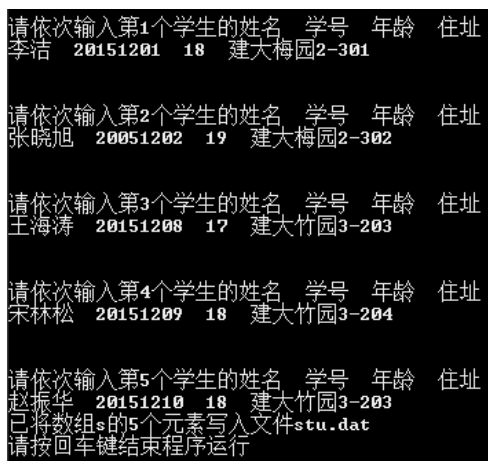
```

int i;
for(i=0; i<N; i++)
{printf("\n\n 请依次输入第%d个学生的姓名 学号 年龄 住址\n" , i+1);
scanf("%s%d%d%s", s[i].name, &s[i].num, &s[i].age , s[i].addr);
}
p=s; /*指针变量p 指向结构体数组s 首地址*/
/*以下语句调用函数 save 将结构体数组s 的N个元素写入文件 stu.dat*/
save( p , N);
printf("已将数组s 的%d个元素写入文件 stu.dat\n" , N);
printf("请按回车键结束程序运行\n");
getchar( );
getchar( );
}

```

运行结果如图 2-33 所示。

思考：以上程序中为什么要有两个 `getchar();` 语句？



```

请依次输入第1个学生的姓名 学号 年龄 住址
李洁 20151201 18 建大梅园2-301

请依次输入第2个学生的姓名 学号 年龄 住址
张晓旭 20051202 19 建大梅园2-302

请依次输入第3个学生的姓名 学号 年龄 住址
王海涛 20151208 17 建大竹园3-203

请依次输入第4个学生的姓名 学号 年龄 住址
宋林松 20151209 18 建大竹园3-204

请依次输入第5个学生的姓名 学号 年龄 住址
赵振华 20151210 18 建大竹园3-203
已将数组s的5个元素写入文件stu.dat
请按回车键结束程序运行

```

图 2-33 例 2.62 运行结果

例 2.63 读取由例 2.62 产生的磁盘文件，按年龄由小到大用白底蓝字输出记录。

程序如下：

```

#include <stdio.h>
#include <stdlib.h>
#define N 5
struct student
{char name[10];
int num;
int age;
char addr[30];
};

void sort(struct student s[ ] , int n )
{struct student st;

```

```
FILE *fp;
int i , j;
/*以下语句以读方式打开 stu.dat 文件*/
if ((fp=fopen("stu.dat", "rb"))==NULL)
{printf("无法打开 stu.dat 文件\n");    exit(0) ;}
for (i=0; i<n; i++)
if (fread(&s[i], sizeof(struct student), 1, fp)!=1)
    {printf("无法从文件 stu.dat 读取数据\n");
    exit(0) ;  }
/*以下双重循环实现按年龄由小到大排序*/
for(j=0; j<(n-1); j++)
    for(i=0; i<(n-1-j); i++)
        if (s[i].age>s[i+1].age)
        {st=s[i];    s[i]=s[i+1];    s[i+1]=st;}
fclose(fp);
}

main()
{struct student s[N] , *p;
int i;
system("cls");
system("color F9"); /*设置白底蓝字*/
p=s; /*指针变量 p 指向结构体数组 s 首地址*/
/*以下语句调用函数 sort 读取文件存放到数组 s 中，按年龄由小到大排序*/
sort( p , N);
printf("%-12s%-12s%-6s%-30s\n","学号","姓名","年龄","住址");
for(i=0; i<N; i++)
printf("%-12d%-12s%-6d%-30s\n",s[i].num, s[i].name, s[i].age, s[i].addr);
}
```

运行结果如图 2-34 所示。

学号	姓名	年龄	住址
20151208	王海涛	17	建大竹园3-203
20151201	李洁	18	建大梅园2-301
20151209	宋林松	18	建大竹园3-204
20151210	赵振华	18	建大竹园3-203
20051202	张晓旭	19	建大梅园2-302

图 2-34 例 2.63 运行结果

例 2.64 设计给小学生随机出两位数加法运算题的强化训练系统。

(1) 主函数 main() 功能

显示菜单，调用以下 func1、func2、func3 三个函数，如图 2-35 所示。

```
两位数加法自动出卷及训练系统
1 随机生成带答案和无答案两种加法试卷
2 输出试卷、自动判卷并生成错题文件
3 输出错题文件内容
4 退出本系统
请输入数字选择以上功能
```

图 2-35 例 2.64 main 函数运行结果

(2) 函数 func1 功能

随机生成两个文本文件：带答案和无答案两种加法试卷，每种试卷内含 20 道题，并且每次运行程序产生的试卷内容都不同，如图 2-36~图 2-38 所示。

随机生成带答案两位数加法试卷 已保存于answer.txt

随机生成无答案两位数加法试卷 已保存于Noanswer.dat
按回车键，返回主菜单程序

图 2-36 例 2.64 func1 函数运行结果

(1) 36+62=
(2) 24+54=
(3) 90+92=
(4) 18+77=
(5) 49+36=
(6) 52+74=
(7) 73+40=
(8) 11+31=
(9) 79+63=
(10) 62+37=
(11) 42+22=
(12) 61+96=
(13) 26+13=
(14) 44+29=
(15) 57+12=
(16) 85+98=
(17) 84+35=
(18) 52+66=
(19) 96+16=
(20) 24+21=

(1) 36+62=98
(2) 24+54=78
(3) 90+92=182
(4) 18+77=95
(5) 49+36=85
(6) 52+74=126
(7) 73+40=113
(8) 11+31=42
(9) 79+63=142
(10) 62+37=99
(11) 42+22=64
(12) 61+96=157
(13) 26+13=39
(14) 44+29=73
(15) 57+12=69
(16) 85+98=183
(17) 84+35=119
(18) 52+66=118
(19) 96+16=112
(20) 24+21=45

图 2-37 例 2.64 func1 函数生成的无答案文件

图 2-38 例 2.64 func1 函数生成的带答案文件

(3) 函数 func2 功能

随机出 20 道两位数相加的运算题，显示在屏幕上，小学生可从键盘上输入计算结果，程序自动判断对错并计分。将小学生做错的题存入一个文本文件中，如图 2-39 所示。

题号	题目	解答	正误	得分
<1 >	78 + 30=108	对	对	5 分
<2 >	96 + 24=120	对	对	5 分
<3 >	87 + 55=132	错	错	0 分
<4 >	44 + 50=94	对	对	5 分
<5 >	62 + 14=76	对	对	5 分
<6 >	86 + 38=124	对	对	5 分
<7 >	62 + 24=86	对	对	5 分
<8 >	87 + 34=121	对	对	5 分
<9 >	64 + 18=82	对	对	5 分
<10>	56 + 17=73	对	对	5 分
<11>	61 + 49=111	错	错	0 分
<12>	62 + 48=110	对	对	5 分
<13>	38 + 40=78	对	对	5 分
<14>	10 + 40=50	对	对	5 分
<15>	45 + 88=132	错	错	0 分
<16>	98 + 40=138	对	对	5 分
<17>	79 + 13=92	对	对	5 分
<18>	93 + 36=129	对	对	5 分
<19>	66 + 31=97	对	对	5 分
<20>	33 + 39=72	对	对	5 分

成绩为85分
已生成错题文件，保存于addderror.txt
按回车键，返回主菜单程序

图 2-39 例 2.64 func2 函数运行结果

(4) 函数 func3 功能

将以上 func2 函数生成的保存错题的文本文件内容显示在屏幕上，如图 2-40 所示。

```
<3> 87+55=
<11> 61+49=
<15> 45+88=
以上是错题文件中的内容，按回车键，返回主菜单程序
```

图 2-40 例 2.64 func3 函数实现屏幕显示

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>
#define N1 21 /*试卷中 20 道题*/
main()
{int k, choice=0;
void func1(int n); /*函数声明*/
int func2(int n);
void func3(int n);
while (choice!=4)
{system("cls");
system("color f9");
printf("\t\t 两位数加法自动出卷及训练系统\n");
printf("\t\t 1 随机生成带答案和无答案两种加法试卷\n");
printf("\t\t 2 输出试卷、自动判卷并生成错题文件 \n");
printf("\t\t 3 输出错题文件内容\n");
printf("\t\t 4 退出本系统\n");
printf("\t\t 请输入数字选择以上功能" );
scanf("%d", &choice);
getchar();
switch (choice)
{case 1: func1(N1); break;
case 2: k=func2(N1); break;
case 3: func3(k); break;
case 4: system("cls"); printf("\t\t 欢迎下次使用本系统\n"); }
}
}

/*以下函数的功能是随机生成两位数加法试卷，分为带答案和无答案两种文件*/
void func1(int n)
{FILE *fp1, *fp2;
int i, sum, m1, m2, k;
long now;
char c1='+', c2='=';
system("cls");
/*以下使用"写"方式打开两个文本文件*/
if ((fp1=fopen("answer.txt", "w"))==NULL)
{printf("无法打开文件\n");
exit(0); }
```



```

if ((fp2=fopen("Noanswer.txt", "w"))==NULL)
    {printf("无法打开文件\n");
    exit(0); }
k=rand();
srand(time(&now)%k);
for(i=1; i<n; i++)
    {do
        m1=rand(); /*生成两位数的随机数*/
        while ( !(m1>=10 && m1<=99) );
        do
            m2=rand();
            while ( !(m2>=10 && m2<=99) );
            sum=m1+m2;
            fprintf(fp1,"(%-2d) %d%c%d%c%d\n",i, m1, c1, m2,c2,sum);
            fprintf(fp2,"(%-2d) %d%c%d%c\n",i, m1, c1, m2,c2);
        }
    fclose(fp1); /*关闭文件 answer.txt*/
    fclose(fp2); /*关闭文件 Noanswer.txt*/
    printf("\n\n\t\t随机生成带答案两位数加法试卷 已保存于 answer.txt\n");
    printf("\n\n\t\t随机生成无答案两位数加法试卷 已保存于 Noanswer.dat\n");
    printf("\t\t按回车键, 返回主菜单程序");
    getchar();
}

#include <windows.h>
void gotoxy(int x, int y) /*此函数将光标定位于 x 列,y 行*/
{COORD pos={x,y};
HANDLE hOut = GetStdHandle(STD_OUTPUT_HANDLE);
SetConsoleCursorPosition(hOut, pos);
}

int func2(int n)
{FILE *fp;
char ci[3], st[N1][15];
int i,k, sum,m1,m2, score=0;
long now;
system("cls");
if ((fp=fopen("adderror.txt", "w"))==NULL)
    {printf("无法打开文件\n");
    exit(0); }
printf("%-6s%-8s%-7s%-8s%-4s\n", "题号", "题目", "解答","正误","得分");
k=rand();
srand(time(&now)%k);
k=0; /*k 统计错题个数*/
for(i=1; i<n; i++)
    {do
        m1=rand(); /*生成两位数的随机数*/

```

```

while ( !(m1>=10 && m1<=99) );
do
m2=rand();
while ( !(m2>=10 && m2<=99) );
gotoxy(1,i+1);
printf("(%-2d)  %d + %d=" ,i, m1, m2);
gotoxy(15,i+1);          /*光标定位于第 i+1 行, 第 15 列*/
scanf("%d" , &sum);
if (sum==m1+m2)
{gotoxy(22,i+1);          /*光标定位于第 i+1 行, 第 22 列 输出"对"*/
printf("%s","对");
gotoxy(30,i+1);          /*光标定位于第 i+1 行, 第 30 列 输出"5 分"*/
printf("%s","5 分\n");
score=score+5;}
else
{k++; /*错题个数加 1*/
gotoxy(22,i+1);
printf("%s","错");
gotoxy(30,i+1);
printf("%s","0 分\n");
strcpy(st[i], "(" ); /*以下 7 个字符串函数形成字符串例如: (1 )78+62= */
strcat(st[i], itoa(i,ci,10) );
strcat(st[i], " " );
strcat(st[i], itoa(m1,ci,10) );
strcat(st[i], "+" );
strcat(st[i], itoa(m2,ci,10) );
strcat(st[i], "=\0" );
fputs(st[i],fp); fputs("\n",fp); } /*将错题写入文件 adderror.txt 中*/
}
printf("\n\n成绩为%d 分\n",score);
if (!(score==100))
printf("已生成错题文件 保存于 adderror.txt\n");
printf("按回车键, 返回主菜单程序");
getchar();
getchar();
fclose(fp);
return (k);          /*返回值为错题个数*/
}

void func3(int n)
{FILE *fp;
char st[N1][18];
int i;
system("cls");
/*以下使用"读"方式打开文本文件*/
if ((fp=fopen("adderror.txt", "r"))==NULL)
{printf("无法打开文件\n");

```

```
    exit(0) ;    }
    for(i=0; i<n; i++)
    {fgets(st[i],strlen(st[i]),fp); /*读取错题文件的内容*/
      strcat(st[i], "\0");          /*末尾加字符串结束标志*/
      printf("%s", st[i]);          /*输出错题*/
    }
    printf("以上是错题文件中的内容，按回车键，返回主菜单程序");
    getchar();  getchar();
    fclose(fp);
}
```

思考：(1) 若以上程序修改为生成有 30 道题的试卷，如何修改？

(2) 函数 func3 中 printf("%s", st[i]); 修改为 puts(s[i]);，屏幕显示有什么不同？

2.11.3 实验项目

1. 实验目的和要求

- (1) 掌握文件、文件指针的概念。
- (2) 掌握打开、关闭、读/写文件等库函数的使用方法。
- (3) 掌握利用 C 语言编写实用程序的方法和技巧。

2. 实验内容和步骤

(1) 从键盘上输入某设计部门 10 人的工号、姓名、年龄、职称存入磁盘文件 filea.dat 中。可参考本章范例程序例 2.62。

(2) 读取、输出上述磁盘文件 filea.dat 中的全部内容，并求 10 人的平均年龄。可参考本章范例程序例 2.63。

(3) 编程：设计给小学生随机出两位数减法运算题的强化训练系统。要求被减数不小于减数。可参考本章范例程序例 2.64。

3. 实验思考和总结

- (1) 解答范例程序或实验项目中的思考题。
- (2) 总结编程调试的经验教训。

第3章

C 语言课程设计

3.1 学生成绩管理系统

3.1.1 软件设计目的

设计学生成绩管理系统旨在训练学生综合运用 C 语言知识的编程能力，了解管理信息系统的开发流程，进一步掌握 C 语言选择结构、循环结构的编程，函数的定义和调用，库函数的正确调用，起泡法排序，结构体数组的定义，结构体数组元素成员的输入、输出，文件的打开、关闭，文件的读操作、写操作的方法。通过学生成绩管理系统的开发、调试为今后开发高质量的信息管理系统打下坚实的基础。

3.1.2 软件结构设计

1. 软件模块结构

学生成绩管理系统由主菜单程序及各功能模块组成。用户在主菜单程序中选择调用不同的功能模块：输入记录模块、浏览记录模块、修改记录模块、查询记录模块、统计成绩模块。各模块详细功能及关系如图 3-1 所示。

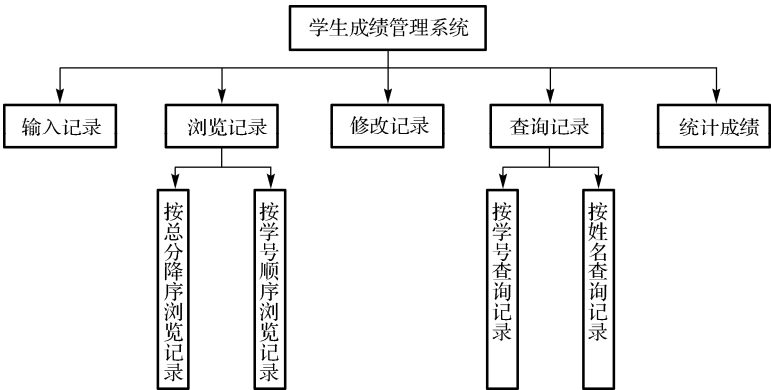


图 3-1 学生成绩管理系统软件模块结构图

2. 软件数据结构

程序中需要定义一个结构体数组 s[N]，N 为符号常量，其数据类型如下：

```
struct student
{long    num;          /*学号*/
char    name[9];       /*姓名*/
int      mark[4] ;}    /*4 个元素依次是数学成绩、英语成绩、C 语言成绩、总成绩*/
```

结构体数组 s[N]的 N 个元素通过运行输入记录的程序模块，可以写入一个二进制文件中。

3.1.3 软件详细设计

1. 主菜单程序的函数原型及功能

(1) 主菜单程序的函数原型

```
main()
```

(2) 主菜单程序的功能

主菜单程序显示出该学生成绩管理系统的所有功能供用户选择，并根据用户选择调用相应的自定义函数。

主菜单程序是一个循环程序，每个功能模块对应的函数被调用后会返回主菜单程序，继续选择其他功能模块，当输入数字 6 时结束主菜单程序的运行。

2. 输入记录程序模块的函数原型及功能

(1) 输入记录程序模块的函数原型

```
void  creates()
```

(2) 输入记录程序模块的功能

输入记录程序从键盘输入 N 个学生的学号，姓名，数学、英语、C 语言 3 科成绩，并计算总分存入结构体数组 s[N]中，将结构体数组各元素逐一写入到学生基本信息二进制文件中。编程思路如图 3-2 所示。

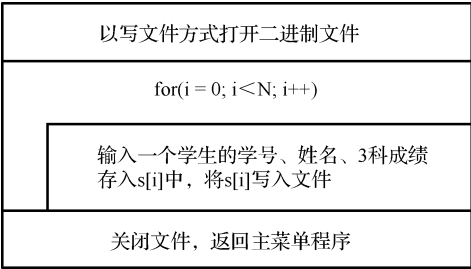


图 3-2 输入记录程序模块 N-S 图

3. 浏览记录程序模块的函数原型及功能

(1) 浏览记录子菜单程序模块的函数原型

```
void  displays()
```

(2) 浏览记录子菜单程序模块的功能

它是一个子菜单程序，可以选择调用按学号顺序浏览记录的程序和按总分降序浏览记录的程序，还可以选择返回主菜单程序。

- 按学号顺序浏览记录程序模块的函数原型

```
void display1()
```

按学号顺序浏览记录程序模块的功能

按学号顺序显示文件中的全部记录，然后返回浏览记录子菜单程序。

- 按总分降序浏览记录程序模块的函数原型

```
void display2()
```

按总分降序浏览记录程序模块的功能

读取文件中的全部记录存放到结构体数组中，然后用起泡法按总分降序排序并显示全部记录，返回浏览记录子菜单程序。

编程思路如图 3-3 所示。

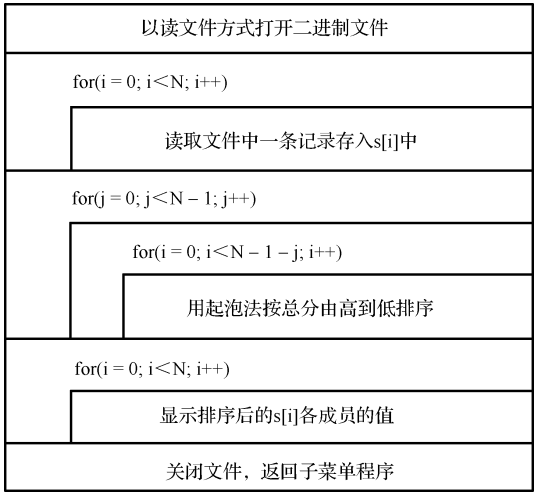


图 3-3 按总分降序浏览记录程序模块 N-S 图

4. 修改记录程序模块的函数原型及功能

- (1) 修改记录程序模块的函数原型

```
void modify()
```

- (2) 修改记录程序模块的功能

该程序可修改多条记录。每次循环用户从键盘输入的可以是一个学生的学号，也可以输入数字 0 作为结束该循环的标志。程序首先判断是否为结束循环的标志 0，若是 0 则结束该循环程序，返回主菜单程序，否则依据学号读取文件中的记录查找相应的待修改记录，然后根据用户的选择修改部分或全部字段并写入文件。输入学号，若查无此人可重新输入。

5. 查询记录程序模块的函数原型及功能

- (1) 查询记录子菜单程序模块的函数原型

```
void search()
```

(2) 查询记录子菜单程序模块的功能

它是一个查询记录子菜单程序，可以选择调用按学号查询记录的程序和按姓名查询记录的程序，还可以选择返回主菜单程序。

● 按学号查询记录程序模块的函数原型

```
void search1()
```

按学号查询记录程序模块的功能

它是按学号可查询多条记录的循环程序，用户从键盘输入学号，程序首先判断用户输入的是否为结束循环的标志 0，若是 0 则结束该循环程序，返回查询子菜单程序，否则依据学号读取文件中的相应记录显示出来，若输入学号有误、查无此人可重新输入。

● 按姓名查询记录程序模块的函数原型

```
void search2()
```

按姓名查询记录程序模块的功能

它是按姓名可查询多条记录的循环程序，用户从键盘输入姓名，程序首先判断用户输入的是否为结束循环的标志 0，若是 0 则结束该循环程序，返回查询子菜单程序，否则依据姓名读取文件中的相应的一条或多条（重名）记录并显示出来，若输入姓名有误、查无此人可重新输入。

6. 统计成绩程序模块的函数原型及功能

(1) 统计成绩程序模块的函数原型

```
statistics()
```

(2) 统计成绩程序模块的功能

它读取存放学生信息的数据文件，统计数学、英语、C 语言这 3 门课成绩的最高分、最低分、平均分，还可以统计 3 门课下列各分数段的人数。

0~59 分、60~69 分、70~79 分、80~89 分、90~100 分。

3.1.4 软件编码设计

1. 主菜单程序的编码设计

```
#define N 39          /*学生人数*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct student
{long num;           /*学号*/
char name[9];        /*姓名*/
int mark[4];         /*4 个元素依次是数学成绩、英语成绩、C 语言成绩、总成绩*/
}
main()
{void creates();      /*函数声明*/
void displays();
void modify();
void search();
void statistics();
int choice=0;
```

```

while (choice!=6)
{system("cls");
  system("color 1f");
  printf("\t\t 学生成绩管理系统\n");
  printf("\t\t 1 输入记录\n");
  printf("\t\t 2 浏览记录\n");
  printf("\t\t 3 修改记录\n");
  printf("\t\t 4 查询记录\n");
  printf("\t\t 5 统计成绩\n");
  printf("\t\t 6 退出本系统\n");
  printf("\t\t 请输入数字选择以上功能" );
  scanf("%d", &choice);
  switch (choice)
  {case 1: creates() ; break; /*调用输入记录的函数 creates()*/
   case 2: displays() ; break; /*调用浏览记录的函数 displays()*/
   case 3: modify() ; break; /*调用修改记录的函数 modify()*/
   case 4: search() ; break; /*调用查询记录的函数 search()*/
   case 5: statistics(); break; /*调用统计成绩的函数 statistics()*/
   case 6: system("cls"); printf("\t\t 欢迎下次使用本系统\n"); } /*结束*/ }
}

```

运行结果如图 3-4 所示。

2. 输入记录程序模块编码设计

/*以下函数功能：输入记录*/

```

void creates()
{struct student s[N];
 FILE *fp; /*定义文件指针*/
 int i;
 system("cls");
 /*以下 if 语句以写方式打开二进制文件*/
 if ((fp=fopen("stu.dat", "wb"))==NULL)
 {printf("无法打开 stu.dat 文件\n");
  exit(-1); }
 for (i=0; i<N; i++)
 {printf("请输入第 %d 个人的学号 姓名 ", i+1);
  scanf("%ld%s", &s[i].num, s[i].name);
  printf("请依次输入数学 英语 C 语言分数 ");
  scanf("%d%d%d", &s[i].mark[0], &s[i].mark[1], &s[i].mark[2]);
  printf("\n");
  if ((i+1)%5==0) system("cls"); /*每输入 5 条记录后清屏*/
  s[i].mark[3]=s[i].mark[0]+s[i].mark[1]+s[i].mark[2]; /*计算总分*/
  /*以下 if 语句实现的是将 s[i] 写入文件 stu.dat*/
  if (fwrite(&s[i], sizeof(struct student), 1, fp)!=1)
  {printf("无法写文件 stu.dat\n");
   exit(-1); }
 }
}

```

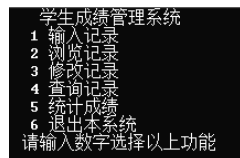


图 3-4 主菜单程序运行结果


```
printf("\n \t\t 已将数据存盘到 stu.dat 文件, 按任意键返回主程序 ");
getchar();
getchar();
fclose(fp);          /*关闭文件*/
}
```

运行结果如图 3-5 所示。

```
请输入第 1 个人的学号 姓名 2015121001 张晓蕊
请依次输入数学 英语 C语言分数 73 65 81

请输入第 2 个人的学号 姓名 2015121002 王刚
请依次输入数学 英语 C语言分数 76 90 78

请输入第 3 个人的学号 姓名 2015121003 李伟
请依次输入数学 英语 C语言分数 69 76 75

请输入第 4 个人的学号 姓名
```

图 3-5 输入记录程序运行结果

3. 浏览记录程序模块的编码设计

(1) 浏览记录子菜单程序

```
/*以下函数功能: 浏览记录子菜单*/
void displays()
{void display1(); /*函数声明*/
void display2();
int choice=0;
while (choice!=3)
{system("cls");
system("color 1f");
printf("\t\t 浏览记录功能: \n");
printf("\t\t 1 按学号顺序浏览记录\n");
printf("\t\t 2 按总分降序浏览记录\n");
printf("\t\t 3 返回主菜单程序\n");
printf("\t\t 请输入数字选择以上功能" );
scanf("%d", &choice);
switch (choice)
{case 1: display1() ; break; /*调用按学号顺序浏览记录的函数 display1( )*/
case 2: display2() ; break; /*调用按总分降序浏览记录的函数 display2( )*/
case 3: return; } /*返回主菜单程序*/
}
}
```

运行结果如图 3-6 所示。

● 按学号顺序浏览记录的程序

```
/*以下函数功能: 按学号顺序浏览记录*/
void display1()
{struct student s[N];
FILE *fp;
int i;
system("cls");
if ((fp=fopen("stu.dat", "rb"))==NULL)
```

```
浏览记录功能:
1 按学号顺序浏览记录
2 按总分降序浏览记录
3 返回主菜单程序
请输入数字选择以上功能
```

图 3-6 浏览记录子菜单程序运行结果

```
{printf("无法打开 stu.dat 文件\n");
exit(-1) ;}
for (i=0; i<N; i++)
fread(&s[i], sizeof(struct student), 1 ,fp);
printf("%-12s%-9s%-9s%-9s%-9s%-9s\n","学号","姓名","数学","英语","C 语言",
      "总分");
for(i=0; i<N; i++)
{printf("%-12ld%-9s",s[i].num, s[i].name);
printf("%-9d%-9d%-9d%-9d\n",s[i].mark[0],s[i].mark[1],s[i].mark[2],
      s[i].mark[3]); }
printf("\n \t\t 按学号顺序显示完毕, 按回车键返回上一级菜单程序");
getchar(); getchar();
fclose(fp);
}
```

运行结果如图 3-7 所示。

学号	姓名	数学	英语	C语言	总分
2015121001	张晓蕊	73	65	81	219
2015121002	王刚	76	90	78	244
2015121003	李伟	69	76	75	220
2015121004	王浩然	63	61	65	189
2015121005	王宝	77	83	82	242
2015121006	赵丽	95	98	96	289
2015121007	李雨	85	91	83	259
2015121008	张雅兰	88	80	81	249
2015121009	王鹏飞	53	58	67	178
2015121010	周晓凤	74	88	70	232
2015121011	张丽萍	87	89	86	262
2015121012	周海涛	55	42	48	145
2015121013	吴靖宇	68	56	54	178
2015121014	宋广鑫	66	87	89	242
2015121015	高剑锋	85	84	73	242
2015121016	张凯	96	99	98	293
2015121017	林志刚	46	75	81	202
2015121018	耿玉亮	78	83	88	249
2015121019	曲振	93	83	62	238
2015121020	徐大鹏	77	80	67	224
2015121021	李俊杰	84	89	92	265
2015121022	郑世斌	75	76	77	228
2015121023	张晓东	86	82	85	253
2015121024	黄海川	56	78	74	208
2015121025	高逸飞	97	94	95	286
2015121026	赵玉龙	87	77	85	249
2015121027	李玉洁	89	73	80	242
2015121028	宋佳佳	96	98	98	292
2015121029	刘志远	88	86	78	252
2015121030	李冉	88	84	85	257
2015121031	姜玲	81	76	66	223
2015121032	王雪纯	64	80	81	225
2015121033	赵伟	79	80	82	241
2015121034	张明凯	95	96	94	285
2015121035	李雨菲	82	70	69	221
2015121036	赵小蒙	77	94	78	249
2015121037	吴文龙	76	88	69	233
2015121038	周志国	86	92	88	266
2015121039	吴文龙	52	83	55	190

按学号顺序显示完毕, 按回车键 返回上一级菜单程序

图 3-7 按学号顺序浏览记录程序运行结果

● 按总分降序浏览记录的程序

```
/*以下函数功能：按总分降序浏览记录*/
void display2()
{struct student s[N] , st;
FILE *fp;
```

```

int i,j;
system("cls");
if ((fp=fopen("stu.dat", "rb"))==NULL)
{printf("无法打开 stu.dat 文件\n");
 exit(-1) ;}
for (i=0; i<N; i++)
fread(&s[i], sizeof(struct student), 1 ,fp); /*从文件中读取一条记录存放到 s[i]中*/

/*以下双重循环程序实现的是用起泡法按总分排序*/
for(j=0; j<(N-1); j++)
for(i=0; i<(N-1-j); i++)
if(s[i].mark[3]<s[i+1].mark[3])
{st=s[i]; s[i]=s[i+1]; s[i+1]=st;}
printf("%-12s%-9s%-9s%-9s%-9s%-9s\n", "学号", "姓名", "数学", "英语", "C 语言",
      "总分");
for(i=0; i<N; i++)
{printf("%-12ld%-9s",s[i].num, s[i].name);
 printf("%-9d%-9d%-9d%-9d\n",s[i].mark[0],s[i].mark[1],s[i].mark[2],
      s[i].mark[3]);
}
printf("\n \t\t\t 按总分降序显示完毕,  按回车键返回上一级菜单程序");
getchar(); getchar();
fclose(fp);
}

```

运行结果如图 3-8 所示。

学号	姓名	数学	英语	C语言	总分
2015121016	张凯	96	99	98	293
2015121028	宋佳佳	96	98	98	292
2015121006	赵丽	95	98	96	289
2015121025	高晓飞	97	94	95	286
2015121034	高明凯	95	96	94	285
2015121038	周志国	86	92	88	266
2015121021	李俊杰	84	89	92	265
2015121011	张丽萍	87	89	86	262
2015121007	李晓雨	85	91	83	259
2015121030	李冉	88	84	85	257
2015121023	张晓东	86	82	85	253
2015121029	刘志远	88	86	78	252
2015121008	张雅兰	88	80	81	249
2015121018	耿王亮	78	83	88	249
2015121026	赵小龙	87	77	85	249
2015121036	赵小蒙	77	94	78	249
2015121002	王刚	76	90	78	244
2015121005	王宝	77	83	82	242
2015121014	宋广磊	66	87	89	242
2015121015	高剑锋	85	84	73	242
2015121027	李玉洁	89	73	80	242
2015121033	赵伟	79	80	82	241
2015121019	曲振	93	83	62	238
2015121037	吴文龙	76	88	69	233
2015121010	周晓岚	74	88	70	232
2015121022	郑世斌	75	76	77	228
2015121032	王雪纯	64	80	81	225
2015121020	徐大	77	80	67	224
2015121031	姜玲	81	76	66	223
2015121035	李雨菲	82	70	69	221
2015121003	李伟	69	76	75	220
2015121001	张晓蕊	73	65	81	219
2015121024	林海川	56	78	74	208
2015121017	林志刚	46	75	81	202
2015121039	吴文龙	52	83	55	190
2015121004	王浩然	63	61	65	189
2015121009	王鹏飞	53	58	67	178
2015121013	吴清宇	68	56	54	178
2015121012	周涛	55	42	48	145

按总分降序显示完毕, 按回车键返回上一级菜单程序

图 3-8 按总分降序浏览记录程序运行结果

4. 修改记录程序模块的编码设计

```
/*以下函数功能：按学号修改记录*/
void modify()
{struct student s[N];
FILE *fp;
int i ,yn;
long num1;
system("cls");
if ((fp=fopen("stu.dat", "rb+"))==NULL)
{printf("无法打开 stu.dat 文件\n");
exit(-1) ; }
while (1)
{printf("\n 请输入需修改的学生记录的学号,当输入学号为 0 时结束修改");
scanf("%ld" , &num1);
if (num1==0) break;
for (i=0; i<N; i++)
{fread(&s[i], sizeof(struct student), 1, fp);
if (s[i].num==num1)
{printf("\n学号 %ld是否修改? 请输入1或2 ,1 修改 2 不修改",s[i].num);
scanf("%d" , &yn);
if(yn==1)
{printf("请输入学生的新学号");
scanf("%ld" , &s[i].num);
}
printf("\n姓名 %s 是否修改? 请输入1或2 ,1 修改 2 不修改", s[i].name);
scanf("%d" , &yn);
if(yn==1)
{printf("请输入学生的新姓名");
scanf("%s" , s[i].name);
}
printf("\n数学 %d 是否修改? 请输入1或2 , 1 修改 2 不修改",s[i].mark[0]);
scanf("%d" , &yn);
if(yn==1)
{printf("请输入学生的新数学成绩");
scanf("%d" , &s[i].mark[0]);
}
printf("\n英语 %d 是否修改? 请输入1或2 , 1 修改 2 不修改" ,s[i].mark[1]);
scanf("%d" , &yn);
if(yn==1)
{printf("请输入学生的新英语成绩");
scanf("%d" , &s[i].mark[1]);
}
}
```

```

printf("\nC语言 %d 是否修改? 请输入1或2, 1 修改 2 不修改", s[i].mark[2]);
scanf("%d", &yn);
if(yn==1)
{
    printf("请输入学生的新C语言成绩");
    scanf("%d", &s[i].mark[2]);
}
s[i].mark[3]=s[i].mark[0]+s[i].mark[1]+s[i].mark[2];
fseek(fp, 0-sizeof(struct student), 1);
if (fwrite(&s[i], sizeof(struct student), 1, fp)!=1)
printf("写文件错误\n");
rewind(fp); break;
}
}
}
fclose(fp);
system("cls");
printf("按回车键可显示修改后的全部记录");
getchar();
getchar();
display1(); /*调用函数 display1(), 显示修改后的全部记录*/
}

```

运行结果如图 3-9、图 3-10 所示。

```

请输入需修改的学生记录的学号.当输入学号为0时结束修改2015121031
学号 2015121031 是否修改? 请输入1或2 .1 修改 2不修改2
姓名 姜玲 是否修改? 请输入1或2 .1 修改 2不修改1
请输入学生的新姓名江玲
数学 81 是否修改? 请输入1或2 . 1 修改 2不修改1
请输入学生的新数学成绩88
英语 76 是否修改? 请输入1或2 . 1 修改 2不修改2
C语言 66 是否修改? 请输入1或2 . 1 修改 2不修改2
请输入需修改的学生记录的学号.当输入学号为0时结束修改2015121003
学号 2015121003 是否修改? 请输入1或2 .1 修改 2不修改2
姓名 李伟 是否修改? 请输入1或2 .1 修改 2不修改2
数学 69 是否修改? 请输入1或2 . 1 修改 2不修改2
英语 76 是否修改? 请输入1或2 . 1 修改 2不修改1
请输入学生的新英语成绩86
C语言 75 是否修改? 请输入1或2 . 1 修改 2不修改2
请输入需修改的学生记录的学号.当输入学号为0时结束修改0

```

图 3-9 修改记录过程中程序运行结果 1

学号	姓名	数学	英语	C语言	总分
2015121001	张晓蕊	73	65	81	219
2015121002	王刚伟	76	90	78	244
2015121003	李浩然	69	86	75	230
2015121004	王浩然	63	61	65	189
2015121005	王宝	77	83	82	242
2015121006	赵丽雨	95	98	96	289
2015121007	李雨兰	85	91	83	259
2015121008	张雅兰	88	80	81	249
2015121009	王鹏飞	53	58	67	178
2015121010	周晓岚	74	88	70	232
2015121011	张丽萍	87	89	86	262
2015121012	周海涛	55	42	48	145
2015121013	吴靖宇	68	56	54	178
2015121014	宋广磊	66	87	89	242
2015121015	高剑锋	85	84	73	242
2015121016	张凯刚	96	99	98	293
2015121017	林志刚	46	75	81	202
2015121018	耿玉亮	78	83	88	249
2015121019	曲振鹏	93	83	62	238
2015121020	徐大木	77	80	67	224
2015121021	李俊斌	84	89	92	265
2015121022	郑世斌	75	76	77	228
2015121023	张晓东	86	82	85	253
2015121024	黄海川	56	78	74	208
2015121025	高逸飞	97	94	95	286
2015121026	赵玉龙	87	77	85	249
2015121027	李玉洁	89	73	80	242
2015121028	宋佳佳	96	98	98	292
2015121029	刘志远	88	86	78	252
2015121030	李冉	88	84	85	257
2015121031	江玲	88	76	66	230
2015121032	王雪纯	64	80	81	225
2015121033	赵伟	79	80	82	241
2015121034	张明凯	95	96	94	285
2015121035	李雨菲	82	70	69	221
2015121036	赵小蒙	77	94	78	249
2015121037	吴文龙	76	88	69	233
2015121038	周志国	86	92	88	266
2015121039	吴文龙	52	83	55	190

按学号顺序显示完毕。按回车键返回上一级菜单程序

图 3-10 修改记录后程序运行结果 2

5. 查询记录程序模块的编码设计

(1) 查询记录子菜单

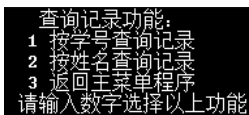
```
/*以下函数功能： 查询记录子菜单*/
void search()
{void search1(); /*函数声明*/
void search2();
int choice=0;
while (choice!=3)
{system("cls");
system("color 1f");
printf("\t\t\t 查询记录功能: \n");
printf("\t\t\t 1 按学号查询记录\n");
printf("\t\t\t 2 按姓名查询记录\n");
printf("\t\t\t 3 返回主菜单程序\n");
printf("\t\t\t 请输入数字选择以上功能" );
scanf("%d", &choice);
```

```

switch (choice)
{case 1: search1() ; break; /*调用按学号查询记录的函数 search1()*/
 case 2: search2() ; break; /*调用按姓名查询记录的函数 search2()*/
 case 3: return; } /*返回主菜单程序*/
}
}

```

运行结果如图 3-11 所示。



```

查询记录功能:
1 按学号查询记录
2 按姓名查询记录
3 返回主菜单程序
请输入数字选择以上功能

```

图 3-11 查询记录子菜单程序运行结果

● 按学号查询记录

```

/*以下函数功能：按学号查询记录*/
void search1()
{struct student s[N];
 FILE *fp;
 int i , j ;
 long num1;
 system("cls");
 if ((fp=fopen("stu.dat", "rb"))==NULL)
 {printf("无法打开 stu.dat 文件\n");
 exit(-1) ; }
 while (1)
 {rewind(fp);
 printf("\n\n 请输入需查询的学生学号,当输入学号为 0 时结束");
 scanf("%ld" , &num1);
 if (num1==0) break;
 for (i=0; i<N; i++)
 {fread(&s[i], sizeof(struct student), 1, fp); /*从文件中读取一条记录
 存放 to s[i]中*/
 if (num1==s[i].num)
 {printf("%-13s%-9s%-8s%-8s%-8s%-8s\n" , "学号", "姓名","数学","英语",
 "C 语言","总分");
 printf("%-13ld%-9s" , s[i].num,s[i].name );
 for (j=0; j<4; j++)
 printf("%-8d" , s[i].mark[j]);
 printf("\n\n 按回车键可继续查询");
 getchar();
 getchar();
 system("cls");
 break;
 }
 }
 if (i==N)

```

```

        { printf("查无此人, 请重新输入学号\n"); continue;}
    }
}
fclose(fp);
}

```

运行结果如图 3-12 所示。



```

请输入需查询的学生学号, 当输入学号为0时结束 2015121025
学号      姓名      数学      英语      C语言      总分
2015121025 高逸飞      97       94       95       286
按回车键可继续查询

```

图 3-12 按学号查询记录程序运行结果

● 按姓名查询记录

```

/*以下函数功能: 按姓名查询记录*/
void search2()
{struct student s[N];
FILE *fp;
int i, j, f;
char name1[9];
system("cls");
if ((fp=fopen("stu.dat", "rb"))==NULL)
{printf("无法打开 stu.dat 文件\n");
exit(-1); }
while (1)
{rewind(fp);
system("cls");
f=0;
printf("\n\n 请输入需查询的学生姓名, 当输入姓名为 0 时结束");
scanf("%s", &name1);
if (strcmp(name1, "0")==0) break;
printf("%-13s%-9s%-8s%-8s%-8s%-8s\n", "学号", "姓名", "数学", "英语", "C 语
言", "总分");
for (i=0; i<N; i++)
{fread(&s[i], sizeof(struct student), 1, fp); /*从文件中读取一条记录存放
到 s[i] 中*/

if (strcmp(name1, s[i].name)==0)
{f=1;
printf("%-13ld%-9s", s[i].num, s[i].name);
for (j=0; j<4; j++)
printf("%-8d", s[i].mark[j]);
printf("\n");
}
}
}
if (f==0)
{printf("查无此人, 请重新输入姓名\n"); continue;}
printf("\n\n 按回车键可继续查询");
}

```



```

    getchar();
    getchar();
    system("cls");
}
fclose(fp);
}

```

运行结果如图 3-13 所示。



图 3-13 按姓名查询记录程序运行结果

6. 统计成绩程序模块的编码设计

```

/*以下函数功能：统计成绩*/
void statistics()
{struct student s[N];
FILE *fp;
int i, j, sum=0;
int score[3][3]={0,0,0},{100,100,100},{0,0,0}};
int count[5][3]={0},{0},{0},{0},{0}};
system("cls");
if ((fp=fopen("stu.dat", "rb"))==NULL)
{printf("无法打开 stu.dat 文件\n");
exit(-1); }
for (i=0; i<N; i++)
fread(&s[i], sizeof(struct student), 1, fp); /*从文件中读取一条记录存
放到 s[i]中*/

/*以下双重循环程序实现的是求 3 门课的最高分、最低分*/
for (i=0; i<N; i++)
for (j=0; j<3; j++)
{if(s[i].mark[j]>score[0][j]) score[0][j]=s[i].mark[j];
else if(s[i].mark[j]<score[1][j]) score[1][j]=s[i].mark[j];
score[2][j]=score[2][j]+s[i].mark[j];
}

/*以下循环程序实现的是求 3 门课的全班平均分*/
for(j=0; j<3; j++)
score[2][j]=score[2][j]/N;

/*以下双重循环程序实现的是统计 3 门课分数及 5 个分数段的人数*/
for (i=0; i<N; i++)
for (j=0; j<3; j++)
if(s[i].mark[j]>=90) count[4][j]++;
else if(s[i].mark[j]>=80) count[3][j]++;
else if(s[i].mark[j]>=70) count[2][j]++;

```

```
        else if(s[i].mark[j]>=60) count[1][j]++;
        else count[0][j]++;
printf("\n\n 各科目分数统计结果如下: \n");
printf("%-10s%-10s%-10s%-10s\n", "统计项目", "数学", "英语", "C 语言");
for(i=0; i<3; i++)
{switch(i)
{case 0: printf("%-10s", "最高分"); break;
 case 1: printf("%-10s", "最低分"); break;
 case 2: printf("%-10s", "平均分"); break;
 }
 for(j=0; j<3; j++)
     printf("%-10d" , score[i][j]);
 printf("\n");
}
printf("\n\n 各分数段人数统计结果如下: \n");
for(i=0; i<5; i++)
{switch(i)
{case 0: printf("%-10s", "0 -59 分"); break;
 case 1: printf("%-10s", "60-69 分"); break;
 case 2: printf("%-10s", "70-79 分"); break;
 case 3: printf("%-10s", "80-89 分"); break;
 case 4: printf("%-10s", "90-100 分"); break;
 }
 for(j=0; j<3; j++)
     printf("%-10d" , count[i][j]);
 printf("\n");
}
printf("\n\n 按回车键返回主菜单程序");
fclose(fp);
getchar();
getchar();
}
```

运行结果如图 3-14 所示。

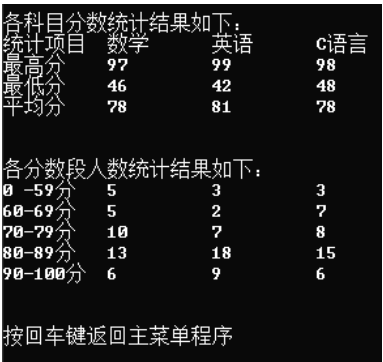


图 3-14 统计成绩程序运行结果

3.2 小学生算术运算试卷自动生成系统

3.2.1 软件设计目的

本程序系统旨在使学生进一步了解管理信息系统的开发流程，提高学生综合运用 C 语言知识编程及调试程序的能力。通过利用 C 语言读、写及产生随机数的一系列函数，设计一个具有实用价值的小学生算术运算试卷自动生成系统。

3.2.2 软件结构设计

小学生算术运算试卷自动生成系统主菜单程序及各功能模块组成如图 3-15 所示。用户在主菜单程序中选择调用不同的功能模块生成下列不同的两位数加、减、乘、除运算试卷。

- 随机生成加、减、乘、除运算试卷；
- 随机生成加、减、乘、除运算按比例试卷；
- 随机生成单一运算符试卷。

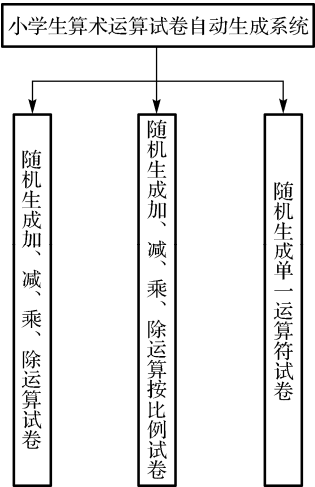


图 3-15 小学生算术运算试卷自动生成系统软件模块结构图

3.2.3 软件详细设计

1. 主菜单程序的函数原型及功能

(1) 主菜单程序的函数原型

main()

(2) 主菜单程序的功能

主菜单程序实现的是显示出小学生算术运算试卷自动生成系统的所有功能供用户选择，并根据用户选择调用相应的自定义函数。

主菜单程序是一个循环程序，每个功能模块对应的函数被调用后就会返回主菜单程序，继续选择其他功能模块，当选择输入数字“4”时结束主菜单程序的运行。

2. 生成指定运算及题量的程序模块函数原型及功能

(1) 生成指定运算及题量的程序模块函数原型

```
void wfile(char bt[],int n,char c[],int k)
```

(2) 生成指定运算及题量的程序模块的功能

形参: bt[]存放一个大题的标题文字, n 存放运算题个数, c[]存放运算符, k 存放一个变化的整数便于产生真正的随机数。它可以生成指定个数和指定运算的两位数算术题存放于文本文件中, 并且每道减法题中被减数大于减数, 每道除法题中被除数能被整除。

3. 随机生成加、减、乘、除运算试卷程序模块的函数原型及功能

(1) 随机生成加、减、乘、除运算试卷程序模块的函数原型

```
void func1()
```

(2) 随机生成加、减、乘、除运算试卷程序模块的功能

本程序模块通过 4 次调用函数 wfile, 生成一个包含两位数加、减、乘、除运算各 25 道题的文本文件。

4. 随机生成加、减、乘、除运算按比例试卷程序模块的函数原型及功能

(1) 随机生成加、减、乘、除运算按比例试卷程序模块的函数原型

```
void func2()
```

(2) 随机生成加、减、乘、除运算按比例试卷程序模块的功能

本程序模块通过 4 次调用函数 wfile, 生成一个包含两位数加、减、乘、除运算试题的文本文件。其中加、减法运算各 30 道题, 乘、除法运算各 10 道题。

5. 随机生成单一运算符试卷程序模块的函数原型及功能

(1) 随机生成单一运算符试卷程序模块的函数原型

```
void func3()
```

(2) 随机生成单一运算符试卷程序模块的功能

本程序模块调用函数 wfile, 可根据用户的选择分别生成加法试卷、减法试卷、乘法试卷、除法试卷。每份试卷包含 100 道题。

3.2.4 软件编码设计

1. 主菜单程序的编码设计

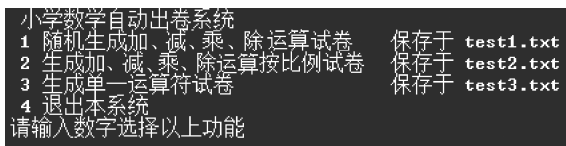
```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>
main()
{int choice=0;
void func1( ); /*函数声明*/
void func2();
void func3();
while (choice!=4)
{system("cls");
```

```

system("color 1f");          /*屏幕显示设置为蓝底白字*/
printf("\t\t 小学数学自动出卷系统\n");
printf("\t\t 1 随机生成加、减、乘、除运算试卷 保存于 test1.txt \n");
printf("\t\t 2 生成加、减、乘、除运算按比例试卷 保存于 test2.txt \n");
printf("\t\t 3 生成单一运算符试卷 保存于 test3.txt \n");
printf("\t\t 4 退出本系统\n");
printf("\t\t 请输入数字选择以上功能" );
scanf("%d", &choice);
getchar();
switch (choice)
{
    case 1: func1(); break;
    case 2: func2(); break;
    case 3: func3(); break;
    case 4: system("cls"); printf("\t\t 欢迎下次使用本系统\n"); }
}
}

```

主菜单程序运行结果如图 3-16 所示。



```

小学数学自动出卷系统
1 随机生成加、减、乘、除运算试卷 保存于 test1.txt
2 生成加、减、乘、除运算按比例试卷 保存于 test2.txt
3 生成单一运算符试卷 保存于 test3.txt
4 退出本系统
请输入数字选择以上功能

```

图 3-16 主菜单程序运行结果

2. 生成指定运算及题量的程序模块的编码设计

```

FILE    *fp; /*全局变量 fp 文件类型指针 */

/*以下函数功能：生成 n 道运算题，运算符为 c*/
/*bt 存放一个大题的标题文字，k 的变化可生成不同的随机数*/
void wfile(char bt[ ], int n, char c[ ], int k)
{
    int n1, n2;          /* n1, n2 是参加算术运算的两个操作数*/
    int i, t;
    long now;
    srand(time(&now)%k); /*确保利用 rand() 生成真正的随机数*/
    fprintf(fp, "%s\n", bt);
    for(i=1; i<=n; i++)
    {
        do
        {
            n1=rand();
            while ( !(n1>=10 && n1<=99) );
        }
        do
        {
            n2=rand();
            while ( !(n2>=10 && n2<=99) );
        }
        if(strcmp(c, "-")==0)
        {
            if (n1<n2) { t=n1; n1=n2; n2=t; } } /*确保被减数大于减数*/
        else if ( strcmp(c, "+")==0)
        {
            do
            {
                n1=rand();

```


3.3 课程设计项目

3.3.1 家庭药品管理系统

1. 数据结构

关于药品基本信息的结构体类型如下：

```
struct drugs
{
    char    num[5];
    char    name[21];
    char    effect[31];
    char    category[10];
    int     stock;
    char    date[11];
    char    location[21];
    char    factory [25];
};
```

以上各结构体成员含义如下：

num[5]	药品编号
name[21]	药名
effect[31]	药品的功能主治
category[10]	类别。分为3类：内服、外用、医疗器具
stock	药品的数量
date[11]	药品的有效截止日期。日期一律用10个字符表示，例如：2018.03.26
location[21]	药品的存放位置
factory [25]	药品的生产厂家

在程序中定义 `struct drugs` 类型的结构体数组，并且写入存放药品基本信息的文件。

2. 功能描述

(1) 主菜单程序模块

家庭药品管理系统由主菜单程序及各功能模块组成。用户在主菜单程序中选择调用不同的功能模块：添加记录模块、浏览记录模块、更新记录模块、查询记录模块、统计记录模块。每个模块执行后返回主菜单程序。

(2) 添加记录模块

用户从键盘上输入药品基本信息，将其存放到一个文件中。

对于同名药品，若药品的功能主治、药品的有效截止日期、药品的生产厂家有任何一项不同，则输入不同的药品编号。

(3) 浏览记录模块

用户可选择调用如下模块：

浏览全部记录模块；
按药品有效期排序浏览记录模块；
按指定类别浏览记录：浏览内服药品记录、浏览外用药品记录、浏览医疗器具记录；
按指定日期浏览过期药品，以便于使用删除记录模块删除过期药品。
(4) 更新记录模块
可调用两个模块：修改记录模块、删除记录模块。
用户从键盘输入药品编号可修改记录、删除记录。
(5) 查询记录模块
可调用两个模块：按药品编号查询记录，按药名查询记录。
(6) 统计记录模块
分别统计内服、外用、医疗器具的数量。

3. 课程设计要求

编程并调试。建立一个 Word 文档且包括如下内容：
(1) 绘制系统功能模块图。
(2) 全部程序清单。
(3) 各程序运行结果截图。
各程序运行结果截图穿插在对应的程序清单下方。
(4) 程序使用说明。

3.3.2 工资管理系统

1. 数据结构

(1) 个人所得税的计算
设计数据结构之前需要了解怎样计算个人所得税。个人所得税的税率表见表 3-1。

表 3-1 个人所得税的税率表

级数	应纳税所得额（含税）	税率（%）	速算扣除数
1	不超过 1500 元	3	0
2	超过 1500 元至 4500 元	10	105
3	超过 4500 元至 9000 元	20	555
4	超过 9000 元至 35 000 元	25	1005
5	超过 35000 元至 55 000 元	30	2775
6	超过 55 000 元至 80 000 元	35	5505
7	超过 80 000 元	45	13 505

全月应纳税所得额=（应发工资-各项社会保险费-其他扣除）-3500（起征点）
说明：如果计算的对象是港澳台人士及外籍人士，则个税起征点应为 4800 元。
缴税=全月应纳税所得额×税率-速算扣除数

(2) 数据结构

```
struct salary
```

```
{int    month;
char   num[6];
char   name[10];
int     category;
float   income[8];
float   deduct[7];
};
```

在程序中定义 `struct salary` 类型的结构体数组，建立存放职工的工资基本信息文件。
以上结构体类型各成员的说明见表 3-2。

表 3-2 结构体类型各成员说明

成员	成员说明	备注
month	月份	
num[6]	工资编号	
name[10]	姓名	
category	起征点类别	1 起征点 3500 元 2 起征点 4800 元
income[0]	岗位工资	
income[1]	薪级工资	
income[2]	基础绩效	
income[3]	奖励绩效	
income[4]	补贴 1	房补
income[5]	补贴 2	交通补贴
income[6]	应发工资	income[0]~income[5]的 6 项之和
income[7]	实发工资	income[7]=income[6]-deduct 数组全部元素
deduct[0]	扣除 1	请假、旷工的工资扣除
deduct[1]	扣除 2	慈善捐款的扣除
deduct[2]	养老保险	
deduct[3]	医疗保险	
deduct[4]	失业保险	
deduct[5]	公积金	
deduct[6]	个人所得税	

2. 功能描述

(1) 主菜单程序模块

工资管理系统由主菜单程序及各功能模块组成。用户在主菜单程序中选择调用不同的功能模块：添加记录模块、浏览记录模块、更新记录模块、查询记录模块、统计记录模块。

每个模块执行后返回主菜单程序。

(2) 添加记录模块

用户从键盘上输入工资基本信息，将其添加到一个文件中。

(3) 浏览记录模块

用户可选择调用如下模块：浏览全部记录模块；按实发工资降序浏览记录模块。

(4) 更新记录模块

可调用两个模块：按工资编号修改记录模块、按工资编号删除记录模块。

(5) 查询记录模块

按工资编号查询记录。

(6) 统计记录模块

统计指定月份起征点为 3500 元和 4800 元的人数。统计指定月份实发工资 1 万元以上的人数。

3. 课程设计要求

编程并调试。建立一个 Word 文档且包括如下内容：

(1) 绘制系统功能模块图。

(2) 全部程序清单。

(3) 各程序运行结果截图。

各程序运行结果截图穿插在对应的程序清单下方。

(4) 程序使用说明。

3.3.3 小学生算术运算题自动批改系统

1. 随机数及文件的要求

要求生成真正的两位数的随机数，即每次运行程序时屏幕上输出不同的运算题。可参考第 2 章例 2.64。用户做错的题存放到文本文件中。

2. 功能描述

小学生算术运算题自动批改系统可在屏幕上输出 20 道加、减、乘、除运算题，用户输入运算结果后，程序可自动批改、计分并在屏幕上输出，同时把用户做错的题存放到文本文件中，可以将错题文件的内容输出以便让用户改错，并自动批改。

(1) 主菜单程序模块

系统由主菜单程序及各功能模块组成。用户在主菜单程序中选择调用不同的功能模块：加法训练程序模块、减法训练程序模块、乘法训练程序模块、除法训练程序模块。每个模块执行后返回主菜单程序。

(2) 加法训练程序模块

加法训练程序模块可选择调用两个模块：加法训练程序模块、加法错题重做程序模块。

加法训练程序模块：随机出 20 道两位数相加的运算题，显示在屏幕上，小学生可从键盘上输入计算结果，程序自动判断对错并计分。小学生做错的题存入一个加法错题文本文件中。

错题重做程序模块：读取加法错题文本文件中的错题输出在屏幕上，让用户改错，并自动批改。

(3) 减法训练程序模块

减法训练程序模块可选择调用两个模块：减法训练程序模块、减法错题重做程序模块。

错题重做程序模块：做错的题存入一个减法错题文本文件中。要求随机生成的运算题中被减数大于减数。

(4) 乘法训练程序模块

乘法训练程序模块可选择调用两个模块：乘法训练程序模块、乘法错题重做程序模块。

错题重做程序模块：做错的题存入一个乘法错题文本文件中。

(5) 除法训练程序模块

除法训练程序模块可选择调用两个模块：除法训练程序模块、除法错题重做程序模块。

错题重做程序模块：做错的题存入一个除法错题文本文件中。要求随机生成的运算题中被除数大于除数且商为整数。

3. 课程设计要求

编程并调试。建立一个 Word 文档且包括如下内容：

(1) 绘制系统功能模块图。

(2) 全部程序清单。

(3) 各程序运行结果截图及生成的文本文件。

各程序运行结果截图及生成的文件穿插在对应的程序清单中。

(4) 程序使用说明。

第4章

VIPEXam 考试学习资源 数据库系统的应用

4.1 VIPEXam 考试学习资源数据库简介

由中科软股教育科技（北京）股份有限公司开发的“VIPEXam 网络考试学习资源库”（以下简称 VIPEXam）是一套专门为高等院校开发的集日常学习、考前练习、在线无纸化考试等功能于一体的教育资源库软件。该系统采用开放、动态的系统架构，将传统的考试、练习模式与先进的网络应用相结合，可使学生根据个性化需求来进行有针对性的学习和考前练习。同时，学校教学部门也可根据需要利用 VIPEXam 构建题库、组织在线考试等，从而快速构建考试和教学平台系统，实现对教学、考试、学习的高效管理。

VIPEXam 题库目前涵盖了英语类、考研类、计算机类、司法类和公务员类等 12 大专辑 1500 余个考试科目的历年真题试卷、强化练习试卷和全真模拟试卷，总题量已超过 14 万套。其中 C 语言专辑收录了 C 语言二级考试笔试、机试的试卷已超过 1500 套，使 VIPEXam 系统成为真正意义上的学习资源数据库。

4.2 C 语言二级考试笔试和机试考试数据库系统的应用

1. 进入系统首页及注册

访问本校图书馆主页，进入 VIPEXam 考试学习资源数据库系统，如图 4-1 所示。

单击“进入数据库”按钮进入 VIPEXam 考试学习资源数据库系统操作界面的首页（后续简称为数据库系统首页），如图 4-2 所示。第一次进入系统要完成个人注册，然后用所注册的用户名和密码进行登录。

2. C 语言专辑的使用

（1）选择及打开试卷

单击“进入数据库”按钮，进入数据库系统首页→单击“模拟自测”菜单→单击左侧专

辑导航的计算机专辑，在国家二级子专辑下有 C 语言专辑，其中又包括机试、机试（改革前）、笔试（改革前）3 大类试卷，如图 4-3 所示。



图 4-1 VIPEXam 考试学习资源数据库登录界面



图 4-2 VIPEXam 考试学习资源数据库系统操作界面的首页

从2013年9月开始,全国计算机等级考试改革为无纸化考试,将原有笔试部分移至上机考试(以选择题形式),考生登录系统后,需要同时完成选择题和操作题的作答。因此在C语言专辑中分为上述3大类试卷。

专辑导航页面上方可提供两个检索文本框,用户可以按科目或按试卷名称进行检索,也可以单击“真题试卷”、“模拟试卷”、“全部试卷”,选择相应的试卷进行测试,如图4-4所示。

例如,选择“2012年9月国家二级(C语言)笔试真题试卷”→单击“查看”→单击下方的“打开试卷”按钮进入介绍该试卷答题时限、试卷难度等基本信息的页面→单击“打开试卷”按钮进入试卷页面→单击“开始答题”按钮系统自动计时,用户便可以答题了,如图4-5所示。



图 4-3 专辑导航



图 4-4 试卷类型选择选项卡

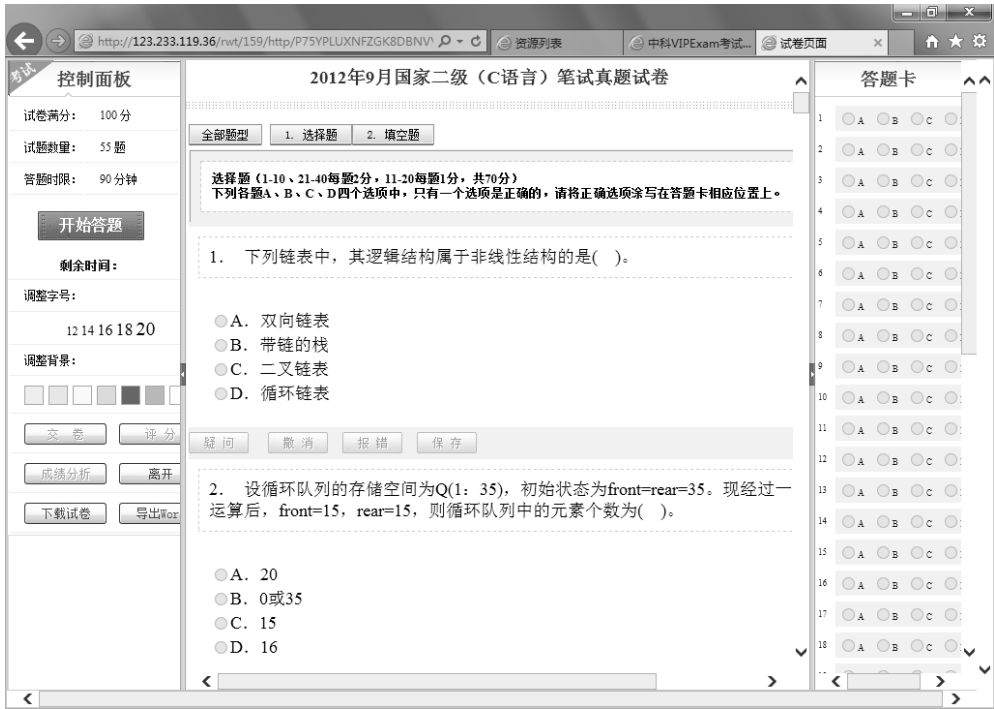


图 4-5 用户选择的 C 语言笔试真题试卷

(2) 交卷

用户答题完毕后，单击“交卷”按钮完成交卷→单击“评分”按钮，系统将自动显示用户每道题的答案、标准答案及解析，如图 4-6 所示。

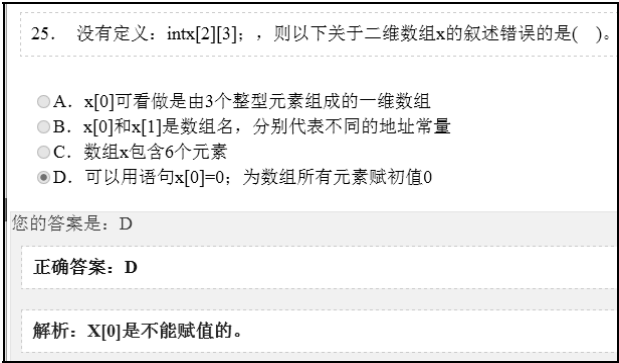


图 4-6 标准答案及解析

(3) 下载试卷

单击“下载试卷”按钮，可将试卷保存为网页文件。若使用 IE 浏览器，要求为 IE 8.0 以上版本。

(4) 导出 Word

若要已将打开的试卷导出到一个 Word 文档中，则单击“导出 Word”按钮。**注意：进入 VIPEXam 考试学习资源数据库首页时，登录教师账号才具有使用该功能的权限。**

4.3 VIPEXam 考试学习资源数据库系统的高级应用

使用 VIPEXam 数据库，用户不仅可以针对各考试科目的历年真题试卷和模拟试卷进行自测，还可以借助许多其他高级功能。

1. 专项练习

系统可根据用户要求在题库中随机抽取任意考试类别、任意题型的试题，以强化训练自己针对某一题型的解题能力。

例如：用户要针对 C 语言二级考试机试中的“程序修改题”这类题型进行专项练习，那么进入 VIPEXam 考试学习资源数据库首页→单击页面左侧的考试目录，选择考试科目 C 语言“机试”→移动鼠标到首页上方“日常学习”菜单，单击“专项练习”→在右侧的“大题型”下拉列表中选择“程序修改题”→输入试题数量、时间设定、试卷名称→单击“抽取试卷”系统自动生成专项练习试卷，如图 4-7 所示。

2. 随机组卷

随机组卷不是随机抽取一份试卷，而是从海量同类试卷中自动抽取试题随机组成一份试卷供用户练习。

例如：用户要针对 C 语言二级考试机试的试卷进行随机组卷，那么进入 VIPEXam 考试学

习资源数据库首页→单击页面左侧的考试目录，选择考试科目 C 语言“机试”→移动鼠标到首页上方“日常学习”菜单，单击“随机组卷”→输入用户自己命名的试卷名称→单击“进入测试”按钮，用户即可打开这份随机组成的试卷进行练习，如图 4-8 所示。



图 4-7 系统自动生成专项练习试卷界面

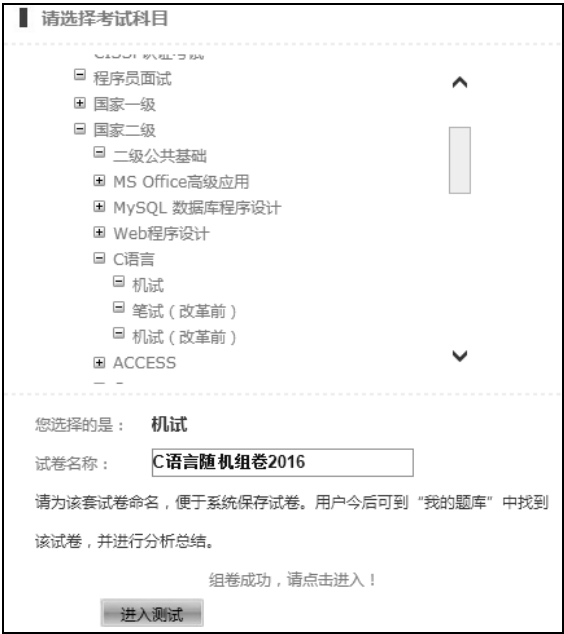


图 4-8 系统随机组卷界面

3. 错题记录

用户登录自己的账号做完题交卷后，系统会自动将试题存入错题记录。

使用方法是：将鼠标移动到首页上方的“错题库”菜单，单击“错题记录”查看用户做错的题，如图 4-9 所示。

试题内容	所属科目	保存日期	操作
，则输出结果是_____。’>有以下程序：#include main() {...	计算机专辑>>国家二级...	2016-05-18	查看 删除
有以下程序：#include main() { int i, s...	计算机专辑>>国家二级...	2016-05-18	查看 删除
若以下选项中的变量全部为整型变量，且已正确定义并赋值...	计算机专辑>>国家二级...	2016-05-18	查看 删除
有以下程序：#include main() } int a=-2...	计算机专辑>>国家二级...	2016-05-18	查看 删除

图 4-9 错题记录查看界面

4. 错题组卷

系统可自动将用户做错的题组成一份试卷。

使用方法是：将鼠标移动到首页上方的“错题库”菜单，选择“错题组卷”→单击“开始组卷”按钮，系统将自动生成一份用户的错题试卷，如图 4-10 所示。

题型名称	试题数量	自定义输入
填空题	2	<input type="text" value="2"/>
选择题（1分）	2	<input type="text" value="2"/>

撤销设定

开始组卷

图 4-10 错题组卷界面

5. 我的题库

用户在“模拟自测”、“专项练习”或“随机组卷”功能下进行自测练习时，可中途退出练习并将试卷保存到“我的题库”中，方便下次登录时重新作答。测试结束后，用户也可将试卷或做错的试题保存到“我的题库”中，以便以后进行自我总结和强化训练。

使用方法是：将鼠标移动到首页上方的“辅助学习”菜单，单击“我的题库”，可显示出用户曾经做过的全部试卷名称清单→单击“选择”按钮可以选中一份试卷→在页面下方可以选择单击“继续答题”按钮、“重新答题”按钮、“删除试卷”按钮，如图 4-11 所示。

2012年9月国家二级（C语言）笔试真题试卷

继续答题

重新答题

删除试卷

图 4-11 “我的题库”操作界面

6. 学习计划

凡事预则立，不预则废。通过“学习计划”功能，用户可以制定自己未来一段时间内的学习计划，以提高学习效率。

使用方法是：将鼠标移动到首页上方的“辅助学习”菜单，单击“学习计划”可显示当月的日历→选择某一天，单击“添加”按钮可输入自己的学习计划，如图 4-12 所示。

<div>2016 年 五月</div> <div>检 索</div>							
星期日	星期一	星期二	星期三	星期四	星期五	星期六	
1 添加	2 添加	3 添加	4 添加 复习考研单词	5 添加	6 添加 C语言二级考试	7 添加	
8 添加	9 添加	10 添加	11 添加	12 添加	13 添加	14 添加	
15 添加	16 添加	17 添加	18 添加	19 添加	20 添加	21 添加	
22 添加	23 添加	24 添加	25 添加	26 添加	27 添加	28 添加	
29 添加	30 添加	31 添加					

图 4-12 输入学习计划界面

7. 单词巩固

系统收录了英语四级、英语六级、考研英语等 10 大类词汇集合，用户可选择其中一种词汇集合系统，按字母 a~z 的顺序排列全部单词，未掌握的单词可收录到难词本中。

使用方法是：将鼠标移动到首页上方的“辅助学习”菜单，单击“单词巩固”→选择未掌握的单词→单击小喇叭图标，可收听该单词的发音→单击“存入难词本”可保存该单词→单击“查看解释”可显示该单词的词义及例句，如图 4-13 所示。

英语四级 英语六级 考研英语 英语三级A级 英语三级B级 专业四级 专业八级 托福考试 雅思考试 GRE考试

考研英语

A开头的单词

B开头的单词

bachelor

backward

bacon

bacterium([pl.]bacteria)

badge

badly

bake

bakery

bald

boundary ['baʊndəri]

存入难词本 查看解释

中文解释:

n. 分界线,边界

例句:

The lane is the boundary of our land.

那条小路就是我们这片地的界线.

The ball ran (ie rolled) to the boundary.

球向着边线滚去.

They marked the boundaries of the football fields.

他们画出了足球场地的边界。

图 4-13 单词巩固操作界面

8. 网络课堂

系统收录了某些考试科目的学习资料供用户下载。

使用方法是：将鼠标移动到首页上方的“学习资料”菜单，单击“网络课堂”→单击首页左侧的专辑导航选择 C 语言专辑，即可显示 C 语言学习资料清单→单击“查看”可打开资料→单击“下载资料”按钮可下载选择的资料，如图 4-14 所示。

资源名称	文件格式	文件大小	发布日期	操作
2016年计算机二级C语言知识点：数据的输入输出	.doc	84KB	2016-02-15	查看
2016年计算机二级C语言知识点：选择结构	.doc	86KB	2016-02-15	查看
2016年计算机二级C语言知识点：C语言基本知识	.doc	95KB	2016-02-15	查看
2013年计算机二级C语言指针考点归纳	.doc	92KB	2013-07-09	查看
2013年计算机二级C语言程序设计基本概念考点归纳	.doc	83KB	2013-07-09	查看
2013年计算机二级C语言中用户标识符的作用域和存储类考点归纳	.doc	82KB	2013-04-18	查看
2013年计算机二级C语言程序设计基本概念考点归纳	.doc	83KB	2013-04-18	查看
2013年计算机二级C语言对函数的进一步讨论考点归纳	.doc	79KB	2013-04-18	查看
2013年计算机二级C语言字符型数据考点归纳	.doc	86KB	2013-04-18	查看
2013年计算机二级C语言循环结构考点归纳	.doc	82KB	2013-04-18	查看

首页

上一页

1

2

3

4

下一页

尾页

共4页

图 4-14 “网络课堂”提供的 C 语言学习资料清单

9. 自建题库

中科 VIPEXam 考试学习资源数据库为图书馆及教师提供了完善而灵活的“自建题库”功能。通过后台“自建题库”功能，图书馆可以灵活、快捷地创建具有本校专业特色的题库，从而更能充分发挥出图书馆在知识传承和衍生创新的过程中，所承担的知识原料供应与催化的服务职能。同时，教师可以通过数据库查找试题编写试卷、组织在线无纸化考试、与学生互动教学，进而方便快捷地构建考试和教学平台系统，实现对教学、考试、学习的高效管理。

10. 数据检索

系统提供了试卷检索、试题检索、试卷全文检索等检索功能。

使用方法是：将鼠标移动到首页上方的“检索导航”菜单，选择“数据检索”→单击“快速检索”选项卡，可选择“试卷级检索”、“试题级检索”、“全文检索”3 种检索方式，如图 4-15 所示。

快速检索

标准检索

高级检索

试卷级检索

检索字段：

试卷名称

关键字：匹配度：

模糊

试卷检索

试题级检索

检索字段：

题干

关键字：匹配度：

模糊

试题检索

全文检索

检索字段：

全文

关键字：匹配度：

模糊

全文检索

图 4-15 快速检索界面

11. 学科导航

系统对于收录的英语类、考研类、计算机类、司法类和公务员类等 12 大专辑 1500 余个考试科目提供了导航功能，图 4-16 仅显示了建筑类部分学科的导航内容。

使用方法是：将鼠标移动到首页上方的“检索导航”菜单，单击“学科导航”。

注册建造师—级
建设工程法规及相关知识 建设工程经济 建设工程项目管理 专业工程管 理与实务
注册建造师二级
建设工程法规及相关知识 建设工程施工管理 专业工程管理与实务
注册造价工程师
工程造价案例分析 工程造价管理基础理论与相关法规 工程造价计价与控 制 建设工程技术与计量（安装） 建设工程技术与计量（土建）
注册监理工程师
建设工程合同管理 建设工程监理案例分析 建设工程监理基本理论与相关 法规 建设工程质量、投资、进度控制
注册设备监理师
设备工程监理基础及相关知识 设备监理合同管理 设备监理综合实务与案 例分析 质量、投资、进度控制
注册城市规划师
城市规划管理与法规 城市规划实务 城市规划相关知识 城市规划原理
注册安全工程师
安全生产法及相关法律知识 安全生产管理知识 安全生产技术 安全生

图 4-16 建筑类部分学科的导航内容

12. 在线考场

中科 VIPExam 数据库的在线无纸化考试功能由前台的“在线考场”和后台的“考试管理”两个功能群组成。

前台的“在线考场”功能群下包含“参加考试”、“成绩查询”两个子功能，是供学生使用的。学生需要使用“考试专用账号”和“密码”才能参加考试。

后台的“考试管理”功能群是供教师使用的。

若在考试过程中发生停电、断网、宕机等特殊情况，学生可向监考老师示意并说明情况。得到监考老师确认后，学生可以重新登录进入考场并继续答卷。学生的全部作答记录都将被自动保存，不会出现已作答记录全部作废的情况。

13. 学习笔记

系统提供了在线建立和管理学习笔记的功能。

使用方法是：将鼠标移动到首页上方的“交互式学习”菜单，单击“学习笔记”→输入文章标题和内容，单击“确定”按钮保存笔记，如图 4-17 所示。

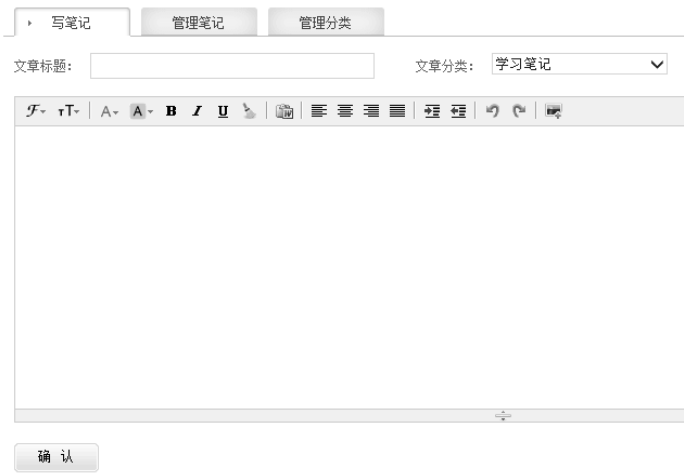


图 4-17 在线建立学习笔记界面

第5章

C 程序的常见错误及调试方法与技巧

5.1 C 程序的常见错误分析

C 语言数据结构、运算符很丰富，语法限制不太严格，设计自由度大。一方面，扩展了 C 语言的功能，增加了 C 语言编程的灵活性。另一方面，也增加了出现错误的概率。初学者了解 C 程序的常见错误及原因，是提高编程和调试程序能力的捷径。

1. 编译预处理命令的相关错误

(1) 在编译预处理命令的行末加分号

```
#include <stdio.h>;
```

错误分析：编译预处理命令不是 C 语句，命令行末不能加分号。

(2) 头文件拼写错误

```
#include <stido.h> /*头文件拼写错误*/
```

出错提示信息：fatal error C1083: Cannot open include file: 'stido.h': No such file or directory

(3) 漏写文件包含编译预处理命令

例 5.1 以下是错误程序，功能是判断 m 是否为素数。

```
#include <stdio.h>
main()
{int m, i, k;
 printf("请输入一个正整数");
 scanf("%d",&m);
 k=sqrt(m);
 for (i=2; i<=k; i++)
 if (m%i==0) break;
 if (i>k) printf("%d 是素数\n",m);
 else printf("%d 不是素数\n",m);
}
```


以上判断 m 是否为素数的程序中调用了求平方根的库函数 `sqrt`，因此，需在 `main` 函数之前添加编译预处理命令：

```
#include <math.h>
```

(4) 定义符号常量错误

```
#define PI=3.14
```

以上定义圆周率符号常量的命令中不能用赋值号，正确的命令如下：

```
#define PI 3.14
```

(5) 对带参宏定义理解错误

初学者容易把带参宏定义当成数学中的函数来分析，例如把以下程序中 `a+b` 的和计算出来再进行宏展开。

例 5.2 以下是正确的程序，实现的是带参宏定义。

```
#define PT 5.5
#define s(x) PT*x*x
main()
{int a=1, b=2;
 printf("%4.1f \n", s(a+b));
}
```

正确的宏展开：`printf("%4.1f\n", 5.5*1+2*1+2);` 运行结果 9.5

错误的宏展开：`printf("%4.1f\n", 5.5*3*3);` 运行结果误认为 49.5

(6) 开发环境设置错误

若 `main` 函数与它调用的各被调函数分别存放在不同的源程序文件中，那么应使用文件包含的编译预处理命令使它们合并在一起，程序如下。

例 5.3 以下是正确的程序，功能是将 `main` 函数与 3 个被调函数分别存放在不同的源程序文件中。

```
/*STUMAIN.C 主菜单程序*/
#define K 10
#include <stdio.h>
#include <stdlib.h>
#include <FCREAT.C> /*被调函数 creates( ) 的文件名 FCREAT.C*/
#include <FDISP.C>
#include <FCHOICE.C>
main()
{int n=0;
 while (n!=4)
 {system("cls");
 system("color 1f");
 printf("\t\t 学生成绩管理系统\n");
 printf("\t\t 1 建立学生成绩数据文件\n");
 printf("\t\t 2 按总分由高到低排序显示各科成绩\n");
 printf("\t\t 3 选择总分 240 分以上的建立另一个学生成绩数据文件\n");
 printf("\t\t 4 退出本系统\n");
 printf("\t\t 请输入数字选择以上功能" );
```

```
scanf("%d", &n);
switch (n)
{
    case 1: creates(); break;
    case 2: displays(); break;
    case 3: choices(); break;
    case 4: system("cls"); printf("\t\t 欢迎下次使用本系统\n"); }
}
```

以上程序在编译时会首先显示错误提示信息：

fatal error C1083: Cannot open include file: 'FCREAT.C': No such file or directory

以上错误是 VC++ 6.0 编译系统找不到被调函数 FCREAT.C 所在的文件夹所致。因此，在编译之前需对 VC++ 6.0 开发环境进行设置。例如：FCREAT.C 所在的文件夹为 D 盘 PRG1032，操作如下：

单击“工具”菜单→选择“选项”命令→单击“目录”选项卡→在路径文本框中输入 D:\PRG1032，如图 5-1 所示。



图 5-1 设置包含的 C 程序文件所在文件夹

2. 与数据类型相关的错误

(1) 数据类型使用不当导致的错误

例 5.4 以下是错误的程序，功能是求 $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{10}$ 之和。

```
#include <stdio.h>
main()
{
    int i, sum=0;
    for(i=1; i<=10; i++)
        sum=sum+1/i;
    printf("sum=%d\n" , sum);
}
```

显然以上数列之和应含有小数，将 sum 定义为整型变量是错误的。

在语句 sum=sum+1/i; 中当 i≥2 时，因为 i 为整型变量，所以使 1/i 的值均为 0，与原意不符。

(2) 忽略了两个整型变量相除结果为整型数

例 5.5 以下是错误的程序。有 36 块砖，由 36 个人来搬，男人搬 4 块，女人搬 3 块，小孩 2 人搬 1 块。要求一次搬完，问男人、女人、小孩各多少人？

```
#include <stdio.h>
main()
{int x,y,z;
 printf("%6s%6s%6s\n", "男","女","小孩");
 for(x=1; x<9; x++)
  for(y=1; y<12; y++)
   {z=36-x-y;
    if (4*x+3*y+z/2==36)
     printf("%6d%6d%6d\n", x,y,z);
   }
}
```

男	女	小孩
1	6	29
3	3	30

运行结果如图 5-2 所示。

图 5-2 例 5.5 运行结果

运行结果中第 1 行小孩人数为 29 显然是错误的，根据原意小孩人数必须为偶数。出错的原因在于 `if (4*x+3*y+z/2==36)` 中 `z` 的值为 29 时，`z/2` 值也是整数 14 而不是 14.5。因此，正确的语句如下：

```
if (4*x+3*y+z/2.0==36)
    printf("%6d%6d%6d\n", x,y,z);
```

修改后 `z` 的值为 29 时，`z/2.0` 值是 14.5 就不符合 `4*x+3*y+z/2.0==36` 的条件了。

(3) 定义变量的数据类型不当导致数据溢出错误

例 5.6 以下是错误的程序，目的是求 1~15 的阶乘。

```
#include <stdio.h>
main()
{int i;
 long t=1;
 for (i=1; i<=15; i++)
 {t=t*i;
  printf("%d!=%ld\n", i,t); }
}
```

```
1!=1
2!=2
3!=6
4!=24
5!=120
6!=720
7!=5040
8!=40320
9!=362880
10!=3628800
11!=39916800
12!=479001600
13!=1932053504
14!=1278945280
15!=2004310016
```

从图 5-3 可以看出由于变量 `t` 发生了数据溢出，导致运行结果 13 的阶乘大于 14 的阶乘。

例 5.7 以下是正确的程序，目的是求 1~15 的阶乘。

图 5-3 例 5.6 错误的运行结果

```
#include <stdio.h>
main()
{int i;
 double t=1;
 for (i=1; i<=15; i++)
 {t=t*i;
  printf("%d!=%.0f\n", i,t);
 }
}
```

运行结果如图 5-4 所示。

```
1!=1
2!=2
3!=6
4!=24
5!=120
6!=720
7!=5040
8!=40320
9!=362880
10!=3628800
11!=39916800
12!=479001600
13!=6227020800
14!=87178291200
15!=1307674368000
```

图 5-4 例 5.7 正确的运行结果

(4) 忽略了变量的类型，进行了不合法的运算

```
#include <stdio.h>
main()
{float a=10, b=6;
 printf("%d", a%b);
}
```

错误分析：%是求余运算符，整型变量 a 和 b 可以进行求余运算，而实型变量不能进行“求余”运算。

3. 与 scanf、printf 库函数相关的错误

在使用 scanf、printf 库函数时初学者常犯如下错误。

(1) 格式声明与变量数据类型不匹配

例 5.8 格式声明与变量数据类型不匹配的错误程序。

```
#include <stdio.h>
main()
{char k1;
 int k2;
 long k3;
 float k4;
 double k5;
 char k6[10];
 printf("请依次输入变量 K1, K2 K3 K4 k5 的值");
 scanf("%c%d%d%f%f" , &k1, &k2, &k3, &k4, &k5); /*错误语句*/
 printf("%c %d %d %f %f\n" , k1, k2, k3, k4, k5); /*错误语句*/
 printf("请输入一个字符串");
 scanf("%s" , &k6); /*错误语句*/
 printf("字符数组 k5 %c\n" , k6); /*错误语句*/
}
```

虽然以上程序中所有输入、输出变量值的语句均有错，但是编译、链接都没有出错提示信息。因此，运行结果如图 5-5 所示。

[illegible]

图 5-5 例 5.8 的运行结果

例 5.9 例 5.8 修改后得到的正确程序。

```
#include <stdio.h>

main()

{char    k1;
  int     k2;
  long    k3;
  float   k4;
  double  k5;
  char    k6[10];

  printf("请依次输入变量 K1, K2 K3 K4 k5 的值");
  scanf("%c%d%ld%f%lf" , &k1, &k2, &k3, &k4, &k5);
  printf("%c %d %ld %f %f\n" , k1, k2, k3, k4, k5);
  printf("请输入一个字符串");
  scanf("%s" , k6);
  printf("字符数组 k5  %s\n" , k6);
}
```

运行结果如图 5-6 所示。

```
请依次输入变量K1, K2 K3 K4 k5的值a 75 8632 16.35 28.593
a 75 8632 16.350000 28.593000
请输入一个字符串apple
字符串数组k5 apple
```

图 5-6 例 5.9 的运行结果

（2）数据输入方式错误

初学者使用库函数 `scanf` 时常因数据输入方式错误导致运行结果错误。数据输入方式正误对照见表 5-1。在 `scanf` 库函数调用方式中最好不出现普通字符，因为需要用户原样输入。

表 5-1 数据输入方式正误对照表

scanf 调用方式	正确输入方式	错误输入方式	说明
int a1, a2, a3; scanf("%d%d%d", &a1, &a2, &a3);	85 72 93	85,72,93	数据之间用空格隔开
int a1, a2, a3; scanf("%d,%d,%d", &a1, &a2, &a3);	85,72,93	85 72 93	数据之间用逗号隔开
int a1, a2; scanf("a1=%d a2=%d", &a1, &a2);	a1=85 a2=72	85 72	普通字符原样输入
int a1, a2; scanf("%d%d\n", &a1, &a2);	85 72\n	85 72	普通字符原样输入
char c1, c2, c3; scanf("%c%c%c", &c1, &c2, &c3);	abc	a b c	输入的字符数据之间不能用空格隔开

(3) 输入库函数选用不当的错误

若输入的字符串中含有空格，那么不能用 scanf 库函数而应选用 gets 库函数。

```
char s1[50] , s2[50];
printf("请输入一个英文语句");
```

```
scanf("%s" , s1);
gets(s2);
```

若输入字符串“this is a book”，则数组 s1 中仅存放了字符串“this”，而 s2 中存放了字符串“this is a book”完整的句子。

(4) 输出函数使用不当的错误

```
char grade;
printf(" grade='A' " );      /*错误。双引号中的内容原样输出，不能为变量赋值*/
printf("%c" , grade='A' ); /*正确。可以为变量赋值并输出*/
```

(5) 输出数据之间无间隔的错误

```
int i, a[10];
...
for(i=0; i<10; i++)
    printf("%d" , a[i]);
```

以上循环输出的 10 个数组元素值输出在同一行，但数据之间无分隔符，因此全部连在了一起。正确的语句如下：

```
for(i=0; i<10; i++)
    printf("%d  ", a[i]); /*相邻数组元素值之间有 3 个空格*/
```

(6) 错用中文标点符号

例 5.10 错用中文标点符号的例程。

```
#include <stdio.h>
main()
{char s[10];
 printf("请输入学生姓名"); /*此句错用中文分号*/
 gets(s);
 puts(s);
}
```

错用中文标点符号，编译系统出错提示：

error C2018: unknown character '0xa3'

4. 漏掉分号或多加分号的错误

(1) 漏掉分号

例 5.11 漏掉分号的例程。

```
#include <stdio.h>
main()
{int a , b;
 a=1
 b=2;
 printf("%d,%d" , a,b);
}
```

编译时出错提示：

error syntax error : missing ';' before identifier 'b'

error C2146: syntax error : missing ';' before identifier 'printf'

编译时, 编译系统在“a=1”后面没发现分号, 就把下一行“b=2”也作为上一行语句的一部分, 这就会出现语法错误。

(2) 在 if 语句中多加分号

```
if (a>b) ;  
max=a;  
else  
max=b;
```

由于在 if (a>b)后多加了分号, 所以当 a>b 时执行了空语句而不是执行 max=a;语句。

(3) 在 for 语句中多加分号

```
for(i=0; i<10; i++) ;  
printf("%d" , a[i]);
```

由于在 for(i=0; i<10; i++)后多加了分号, 所以循环 10 次仅执行空语句, 而 printf(“%d”, a[i]);不是循环体内的语句了。

5. 书写标识符时, 忽略了大小写字母的区别

例 5.12 以下是错误的程序, 忽略了变量大小写的区别。

```
#include <stdio.h>  
main()  
{int a=5;  
printf("%d" , A);  
}
```

出错提示: error C2065: 'A': undeclared identifier (A 是没有定义的标识符), 编译系统把 a 和 A 认为是两个不同的变量名。

6. 变量未赋值的错误

例 5.13 以下是错误的程序, 目的是求 1~100 之间偶数的累加和。

```
#include <stdio.h>  
main()  
{int i, sum;  
for(i=2; i<=100; i+=2)  
sum=sum+i;  
printf("sum=%d\n" , sum);  
}
```

以上程序在编译、链接时均未报错。但由于变量 sum 未赋初值 0 导致输出结果为负数, 如图 5-7 所示。



sum=-858990910

图 5-7 例 5.13 的运行结果

7. 对于复合语句漏掉大括号的错误

(1) if 语句漏掉大括号的错误

例 5.14 以下是错误的程序，实现的是从键盘输入三角形三条边长，并求面积。

```
#include    <stdio.h>
#include    <math.h>
main()
{float  a, b, c , s, area;
printf("请输入三角形的 3 条边长");
scanf("%f,%f,%f", &a,&b,&c);
if (a+b>c&&b+c>a&&c+a>b)
    s=0.5*(a+b+c);                /*此句前缺少大括号*/
    area=sqrt(s*(s-a)*(s-b)*(s-c));
    printf("a=%.2f  b=%.2f  c=%.2f\n",a,b,c);
    printf("area=%.2f\n",area);    /*此句后缺少大括号*/
else
    printf("不能构成三角形\n");
}
```

编译时出错提示信息：error C2181: illegal else without matching if。由于可以构成三角形时需执行 4 条语句，因此，需用大括号将 4 条语句括起来。正确的程序如下。

例 5.15 以下是正确的程序，实现的是从键盘输入三角形三条边长，并求面积。

```
#include    <stdio.h>
#include    <math.h>
main()
{float  a, b, c , s, area;
printf("请输入三角形的 3 条边长");
scanf("%f,%f,%f", &a,&b,&c);
if (a+b>c&&b+c>a&&c+a>b)
{ s=0.5*(a+b+c);
  area=sqrt(s*(s-a)*(s-b)*(s-c));
  printf("a=%.2f  b=%.2f  c=%.2f\n",a,b,c);
  printf("area=%.2f\n",area);
}
else
    printf("不能构成三角形\n");
}
```

(2) 循环语句漏掉大括号的错误

例 5.16 以下是错误的程序，实现的是求 1~10 之间自然数的累加和及 10 的阶乘。

```
#include    <stdio.h>
main()
{int  i, sum=0;
long  t=1;
for(i=1; i<=10; i++)
    sum=sum+i;                    /*此句前缺少大括号*/
}
```



```

    t=t*i;          /*此句后缺少大括号*/
    printf("sum=%d   t=%ld\n" , sum, t);
}

```

由于循环体语句是两条语句，因此，`sum=sum+i;` 和 `t=t*i;` 语句需用一对大括号括起来。

8. 对转义字符的理解

例 5.17 以下是正确的程序，目的是求字符串的长度。初学者分析下列程序运行结果容易出错。

```

#include <stdio.h>
#include <string.h>
main()
{char s1[ ]="\t\b\\xyz\0pen";
 char s2[ ]="computer";
 char s3[ ]="电脑";
 char s4[ ]="\x69\082\n";
 char s5[ ]="\x69\072\n";
 printf("s1 长度 %d\n",strlen(s1));
 printf("s2 长度 %d\n",strlen(s2));
 printf("s3 长度 %d\n",strlen(s3));
 printf("s4 长度 %d\n",strlen(s4));
 printf("s5 长度 %d\n",strlen(s5));
}

```

s1	长度	6
s2	长度	8
s3	长度	4
s4	长度	1
s5	长度	3

运行结果如图 5-8 所示。字符串“`\x69\082\n`”的长度为 1，其中‘`\x69`’为转义字符，占 1 个字节，后跟字符串结束标志‘`\0`’，因为‘`\082`’不可能是八进制数。

图 5-8 例 5.17 的运行结果

字符串“`\x69\072\n`”的长度为 3，其中 3 个转义字符‘`\x69`’、‘`\072`’、‘`\n`’各占 1 个字节。

9. 小括号不配对的错误

当语句中使用多个小括号时，常因粗心使小括号不配对。

```

if ((fp=fopen("stu.dat", "wb")==NULL) /*少一个右括号*/
{printf("无法打开 stu.dat 文件\n");
 exit(-1) ; }

```

10. 数学符号与 C 语言符号混淆的错误

(1) 等号与赋值号混淆的错误

```

if (4*x+3*y+z/2=36)
    printf("%6d%6d%6d\n", x,y,z);

```

在 C 程序中 `==` 为等号，以上语句应修改为：

```

if (4*x+3*y+z/2==36)
    printf("%6d%6d%6d\n", x,y,z);

```

(2) 大于等于或小于等于的错误

```

if (1≤x≤10) y=3x+1;          /*错误的语句*/
if (x>=1&&x<=10) y=3*x+1;    /*正确的语句*/

```

(3) 赋值语句与数学等式混淆的错误

$x+y+z=36$ 是数学等式，不能作为语句写在程序中，可以写一个与它等价的赋值语句：

$z=36-x-y;$

11. if 语句选择不当的错误

例 5.18 以下是错误的程序，实现的是输入一个百分制的成绩，然后输出成绩及相应的等级。下列程序中 if 语句选用不当，应使用多分支的 if 语句，而程序中只使用了简单的 if 语句。

成绩 score	等级
90 分以上	A
80~89 分	B
70~79 分	C
60~69 分	D
59 分以下	E

```
#include <stdio.h>
main()
{int score;
 char grade ;
 printf("请输入学生成绩 " );
 scanf("%d", &score);
 if (score>=90)      grade='A';
 if (score>=80)      grade='B';
 if (score>=70)      grade='C';
 if (score>=60)      grade='D';
 if (score<60)       grade='E';
 printf("成绩为%d,相应等级是%c\n", score, grade);
}
```

以上程序运行时，即使百分制的成绩 score 在 90 分以上，也要执行 if(score>=80)开始的 4 条 if 语句的判断功能。显然是选用 if 语句错误，应使用多分支的 if 语句。

例 5.19 选用多分支的 if 语句的正确程序。

```
#include <stdio.h>
main()
{int score;
 char grade ;
 printf("请输入学生成绩 " );
 scanf("%d", &score);
 if (score>=90)      grade='A';
 else if (score>=80) grade='B';
 else if (score>=70) grade='C';
 else if (score>=60) grade='D';
 else grade='E';
 printf("成绩为%d,相应等级是%c\n", score, grade);
}
```

12. .switch 语句的各分支中漏掉 break 语句的错误

初学者在使用 switch 语句时容易漏掉各分支中的 break 语句。

```
scanf("%d", &score);
temp=score/10;
switch (temp)
{
    case 10:
        case 9:  grade='A';  break;
        case 8:  grade='B';  break;
        case 7:  grade='C';  break;
        case 6:  grade='D';  break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0:  grade='E';
    }
printf("成绩为%d,相应等级是%c\n", score, grade);
```

若漏掉以上 4 个 break 语句, 无论为百分制成绩 score 输入何值都会输出相应等级是 E。

13. 循环程序常见错误

(1) 循环条件写反了导致不执行循环体

例 5.20 以下是错误程序, 目的是求 $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$ 之和, 当某一项小于 10^{-3} 时停止累加。

```
#include <stdio.h>
main()
{
    int i=1;
    float sum=0;
    while ((1.0/i)<1e-3)
    {
        sum=sum+1.0/i;
        i++;
    }
    printf("sum=%f\n", sum);
}
```

由于 while ((1.0/i)<1e-3) 第 1 次判断就不符合条件, 因此, 循环体未被执行。sum 的值为 0, 与原意不符。

例 5.21 以下是正确的程序, 目的是求 $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$ 之和, 当某一项小于 10^{-3} 时停止累加。

```
#include <stdio.h>
main()
{
    int i=1;
    float sum=0;
    while ((1.0/i)>=1e-3)
```

```

    {sum=sum+1.0/i;
      i++;
    }
    printf("sum=%f\n",sum);
}

```

(2) 循环控制变量的值不改变导致无限循环

例 5.22 以下是错误的程序,目的是求 $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$ 之和,当某一项小于 10^{-3} 时停止累加。

```

#include <stdio.h>
main()
{int i=1;
  float sum=0;
  while ((1.0/i)>=1e-3)
    sum=sum+1.0/i;
  printf("sum=%f\n",sum);
}

```

循环控制变量的 i 值不改变,导致无限循环。

14. 与数组相关的错误

(1) 定义数组错误

定义数组时误用变量。

```

int n=10;
int a[n];

```

数组长度可以是常数和符号常量。不允许对数组的大小作动态定义。

(2) 数组元素赋初值错误

数组元素赋初值不能用以下 2 个语句完成。

```

int a[10];
a[10]={1,2,3,4,5,6,7,8,9,10};

```

正确的赋初值语句如下:

```

int a[10]={1,2,3,4,5,6,7,8,9,10};

```

或

```

int a[ ]={1,2,3,4,5,6,7,8,9,10};

```

(3) 数组下标的相关错误

例 5.23 以下是错误的程序,实现的是对 8 个数组元素按起泡法由小到大排序。

```

#include <stdio.h>
main()
{int a[8], i, j, t;
  printf("input numbers :\n");
  for (i=0; i<=8; i++) /*误将 i<8 写成 i<=8,导致数组下标越界*/
    scanf("%d", &a[i]);
  for(j=0; j<7; i++) /*误将 j++写成 i++*/
    for(i=0 ;i<8-j ;i++) /*误将 7-j 写成 8-j*/

```

```

        if (a[i]>a[i+1]) {t=a[i]; a[i]=a[j+1] ;a[j+1]=t;} /*误将a[i+1]写成a[j+1]*/
printf("the sorted numbers :\n");
for(i=0; i<8; i++)
    printf("%d    ",a[i]);
printf("\n");
}

```

以上程序中罗列了初学者常犯的错误。

例 5.24 以下是正确的程序，实现的是对 8 个数组元素按起泡法由小到大排序。

```

#include <stdio.h>
main()
{int  a[8], i, j, t;
printf("input numbers :\n");
for (i=0; i<8; i++)
    scanf("%d", &a[i]);
for(j=0; j<7; j++)
    for(i=0 ;i<7-j ;i++)
        if (a[i]>a[i+1]) {t=a[i]; a[i]=a[i+1] ;a[i+1]=t;}
printf("the sorted numbers :\n");
for(i=0; i<8; i++)
    printf("%d    ",a[i]);
printf("\n");
}

```

(4) 数组名与指针变量名混淆的错误

例 5.25 错误使用数组名的例程。

```

#include <stdio.h>
main()
{char s[20]; /*定义了字符数组 s[20]*/
s="apple" ; /*错误: 数组名 s 是常量不能为它赋值。*/
printf("%s\n" , s);
}

```

不能为数组名赋值，正确的语句是：char s[20]="apple";

例 5.26 正确使用字符型指针变量的例程。

```

#include <stdio.h>
main()
{ char *s; /*定义了字符指针变量 s*/
s="apple"; /*s 存放字符串"apple"的起始地址，不是存放该字符串本身*/
printf("%s\n" , s); /*输出字符串"apple"，而不是输出 s 的值*/
}

```

对初学者容易造成的理解偏差的语句在程序中已注释。

15. 变量、字符常量和字符串常量混淆的错误

```

char c;
c='c'; /*c 为变量 'c'为字符常量 */

```

`c="c";` /*错误的语句: "c"为字符串常量不能给 `c` 赋值*/

在这里就混淆了字符常量与字符串常量,“c”实际上包含两个字符:‘c’和‘\0’,而字符变量 `c` 只能存放 1 个字符,需要用字符数组存放“c”。

16. 字符串比较与数值变量比较混淆的错误

字符串比较大小不能直接使用关系运算符。以下语句是错误的。

```
if ("box">"book") printf("%s", "box"); /*错误的语句*/
else printf("%s", "book");
```

字符串比较大小需要使用库函数 `strcmp`。正确的比较语句如下:

```
if (strcmp("box","book")>0) printf("%s", "box");
else printf("%s", "book");
```

或

```
char s1[20]="box";
char s2[20]="book";
if (strcmp(s1,s2)>0) puts(s1);
else puts(s2);
```

17. 与函数相关的错误

例 5.27 有函数返回值的错误例程。

```
#include <stdio.h>
main()
{int i;
 float a, b, c, s;
 /*此处漏掉被调函数的声明*/
 for( i=1; i<=3; i++)
 {printf("请输入三个数");
 scanf("%f%f%f", &a, &b, &c);
 s=func(float a, float b); /*错误: 多加了两个 float, 漏掉了实参 c*/
 printf("s=%.2f a= %.2f b =%.2f c =%.2f \n", s, a, b, c);
 }
}

int func(float x, float y, float z, float score); /*错误: 函数首部
                                                    多加了分号*/

{ /*变量 score 应该在函数体中定义*/
 x=0.1*x; y=0.1*y; z=0.8*z;
 score=x+y+z;
 return (score); /*错误: 返回值的类型 float, 与函数首部定义的类型不一致*/
}
```

以上程序中罗列了初学者常犯的错误。

例 5.28 有函数返回值的正确例程。

```

#include    <stdio.h>
main()
{int    i;
  float  a, b ,c , s;
  float  func(float x , float y , float z);
for( i=1; i<=3; i++)
    {printf("请输入三个数");
     scanf("%f%f%f", &a, &b, &c);
     s=func(a , b , c);
     printf("s=%.2f a= %.2f b =%.2f c =%.2f \n", s, a , b ,c);
    }
}

float func(float x , float y , float z)
{float  score;
 x=0.1*x;      y=0.1*y;      z=0.8*z;
 score=x+y+z;
 return (score);
}

```

18. 与指针相关的错误**(1) 指针变量未赋值就引用的错误**

```

float a=35.78, *p;
*p=*p+3;    /*错误：指针变量未赋值不能用*p引用*/
P=&a;
*p=*p+3;    /*正确：指针变量已被赋值为&a，可以引用*/

```

(2) 类型不同的指针混用

```

int    a=5, *p1;
float  b=3.7 , *p2;
p1=&a;
p2=p1; /*p2是指向 float 变量的指针, p1 不能给 p2 赋值*/

```

19. 与结构体类型相关的错误**例 5.29** 以下是错误的程序，实现的是有嵌套定义的结构体变量的成员引用。

```

#include    <stdio.h>
main()
{struct s
{char  name[9];
  char  sex[3];
  struct data
  {int  year;
   int  month;
   int  day;

```

```
    } birthday;
};
struct s={"张莉", "女", 1995,3,26}; /*错误: s 不是结构体变量, 不能赋初值*/
printf("%s  %s  ", stu.name, stu.sex);
printf("%d年", s.year); /* 错误: 不能用 s 引用成员, 而且漏掉了 birthday*/
printf("%d月", s.month);
printf("%d日\n", s.day);
}
```

以上程序中 s 不是结构体变量。在语句 `struct s stu;` 中, `struct s` 是结构体类型名, 下列程序中的 `stu` 是结构体变量。

例 5.30 以下是正确的程序, 实现的是有嵌套定义的结构体变量的成员引用。

```
#include <stdio.h>
main()
{struct s
{char name[9];
char sex[3];
struct data
{int year;
int month;
int day;
} birthday;
};
struct s stu={"张莉", "女", 1995,3,26};
printf("%s  %s  ", stu.name, stu.sex);
printf("%d年", stu.birthday.year);
printf("%d月", stu.birthday.month);
printf("%d日\n", stu.birthday.day);
}
```

20. 与文件读写相关的错误

(1) 使用文件时漏掉打开文件的语句, 或打开方式与读写操作函数不匹配。

例如: 以只读方式打开文件, 却调用写文件的函数对它进行写操作。

(2) 打开文件时, 指定的文件名系统找不到

```
if ((fp=fopen("stu.dat", "rb"))==NULL)
{printf("无法打开 stu.dat 文件\n");
exit(-1) ;}
```

以上语句要求文件 `stu.dat` 必须在当前文件夹中, 否则系统会认为该文件不存在。若需要打开的文件不在当前文件夹中, 应在语句中指定它所在的文件夹。正确的语句如下:

```
if ((fp=fopen("D:\\PRG1032\\stu.dat", "rb"))==NULL)
{printf("无法打开 stu.dat 文件\n");
exit(-1) ;}
```


21. 不可见字符错误

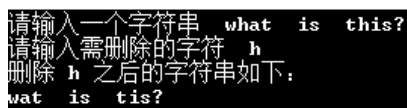
字符串结束标志‘\0’虽然是不可见字符，但在程序中若少了它，则会导致运行错误。

例 5.31 以下是正确的程序，目的是删除一个字符串中指定的字符。

```
#include <stdio.h>
#include <string.h>
main()
{char ch;
 char str[50] , temp[50];
 int i, j=0;
 printf("请输入一个字符串 ");
 gets(str);
 printf("请输入需删除的字符 ");
 ch=getchar( );
 for (i=0; str[i]!='\0'; i++)
     if (str[i]!=ch)
         {temp[j]=str[i];
          j++; }
 temp[j]='\0'; /*在最后一个非删除字符后加上字符串结束标志*/
 strcpy(str , temp);
 printf("删除 %c 之后的字符串如下: \n" , ch );
 puts(str);
}
```

以上程序的运行结果如图 5-9 所示。

若删除 `temp[j]='\0'` 语句，就会出现如图 5-10 所示意想不到的错误提示窗口。



```
请输入一个字符串 what is this?
请输入需删除的字符 h
删除 h 之后的字符串如下:
wat is tis?
```

图 5-9 例 5.31 的运行结果

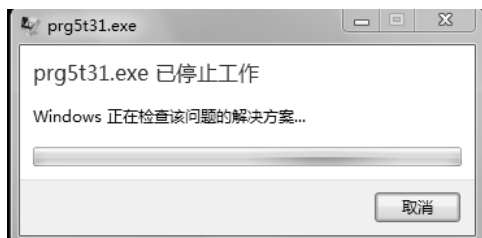


图 5-10 例 5.31 修改后运行时的错误提示窗口

22. 在一个工程中建立了两个 main 函数的错误

若忘记关闭工作区，在一个工程中建立了两个 main 函数，那么链接时会出错。

23. 未关闭运行程序界面的错误

若未关闭正在运行的程序界面，修改该程序之后再次发送“链接”命令，就会显示出错提示信息：

LINK : fatal error LNK1168: cannot open Debug/P1.exe for writing

24. 程序算法不当

程序算法不同于解数学题的方法，初学者往往以为只要运行结果正确就可以了，却忽视了程序算法的合理性。

例 5.32 以下是不合理的算法，目的是对于一个 3×4 的二维数组，从键盘输入整数为全部数组元素赋值，并求所有靠外侧的元素之和。

```
#include <stdio.h>
main()
{int a[3][4], i, j, sum=0;
 printf("请输入数组元素值\n");
 for (i=0; i<3; i++)
     for (j=0; j<4; j++)
         scanf("%d",&a[i][j]);
 printf("\n");
 for (i=0; i<3; i++)
     for (j=0; j<4; j++)
         sum=sum+a[i][j];
 sum=sum-a[1][1]-a[1][2];
 printf("外侧的元素之和 sum=%d\n", sum);
}
```

以上程序算法是先求出全部元素之和，然后减去不是外侧的 2 个元素 $a[1][1]$ 、 $a[1][2]$ 。但是若修改为求 300×400 的二维数组外侧元素之和，其内部元素就不是仅有 2 个了，以上算法就不再适用了。

例 5.33 以下是正确的算法，目的是对于一个 3×4 的二维数组，从键盘输入整数为全部数组元素赋值，并求所有靠外侧的元素之和。

```
#include <stdio.h>
main()
{int a[3][4], i, j, sum=0;
 printf("请输入数组元素值\n");
 for (i=0; i<3; i++)
     for (j=0; j<4; j++)
         scanf("%d",&a[i][j]);
 printf("\n");
 for (i=0; i<3; i++)
     for (j=0; j<4; j++)
         if (i==0 || i==2 || j==0 || j==3) sum=sum+a[i][j];
 printf("外侧的元素之和 sum=%d\n", sum);
}
```

运行结果如图 5-11 所示。



请输入数组元素值
4 2 1 3
8 7 9 5
9 6 7 8
外侧的元素之和 sum=53

图 5-11 例 5.33 的运行结果

25. 程序逻辑错误

程序逻辑错误是指程序无语法错误，也能正常运行，运行结果却与原意不符。这类错误通常是由于程序编写有错或程序算法有错引起的。

(1) 编程错误

```
sum=0;
k=1;
while (k<=100)
    sum=sum+k;
    k++;
```

以上循环语句因漏掉了循环体的大括号，在循环中不执行 `k++`，导致程序无限循环。

(2) 程序算法错误

例 5.34 以下是错误的程序，实现求 100~300 之间的全部素数。

```
#include <stdio.h>
#include <math.h>
main()
{int i, j, count=0;
 for(i=100;i<=300;i++)
  for(j=2;j<=sqrt(i);j++)
   if(i%j==0)
    break;
   else
    {printf("%d 是素数.\n",i);
     count++;
     break;
    }
 printf("总共有素数%d 个",count);
}
```

以上程序由于算法错误，输出的不是 100~300 之间的全部素数而是全部奇数。

例 5.35 以下是正确的程序，实现求 100~300 之间的全部素数。

```
#include <stdio.h>
#include <math.h>
main()
{int n,k,i,count=0;
 for(n=101; n<=300; n=n+2)
  {k=sqrt(n);
   for(i=2;i<=k;i++)
```

```
        if (n%i==0) break;
    if (i>=k+1)
    {printf("%d ",n);
        count++;
        if(count%10==0) printf("\n");
    }
}
printf("\n");
}
```

5.2 C 程序的调试方法

调试程序是通过出错现象找出原因的一种思维分析过程，它不但具有很强的技巧性，而且需要一套科学的调试查错方法。

逻辑错误是初学者感到最棘手的一类错误。因为程序能编译、链接、运行，也无数据输入错误，但得不到正确的运行结果。因此我们借鉴“软件工程”课程中的软件测试和软件调试方法，结合调试经验介绍几种调试方法。

1. 错误推测法

错误推测法：采用逆向思维方式，结合以往测试经验和直觉设计软件在功能和流程上可能存在的各种错误，进行容错性测试。

如将本书第 2 章例 2.25 的程序中删除以下一条语句：

```
if(sales<0) {printf("输入的销售额是负数，请重新输入\n"); continue;}
```

使它修改为缺少容错性处理的错误程序。

例 5.36 以下是错误的程序。设计一个循环程序：某公司销售人员的实发工资由底薪 1000 元和销售额提成两部分组成。具体提成比例如下：

销售额（元）	提成比例
销售额≤3000	无提成
3000<销售额≤6000	7%
6000<销售额≤10 000	10%
销售额>10 000	13%

已知销售额单位为元，是一个整数。用 switch 语句编程，从键盘输入一个销售额，计算该员工的实发工资。当输入的销售额为-1 时结束循环。要求实发工资保留 2 位小数。

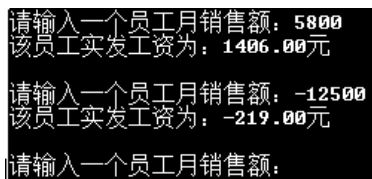
```
#include <stdio.h>
main()
{int n,sales;
float salary=1000;
while (1)
{printf("请输入一个员工月销售额：");
scanf("%d",&sales);
if (sales==-1) break; /*输入结束标志-1 时，跳出循环*/
if (sales%1000==0) /*判断销售额/1000 是否为整数*/
```

```

n=sales/1000;          /*销售额/1000 是整数时, 用此公式求 n 的值*/
else
n=sales/1000+1;        /*销售额/1000 不是整数时, 用此公式求 n 的值*/
switch(n)
{case 0:
 case 1:
 case 2:
 case 3: salary = salary +0; break; /*销售额≤3000 元无提成时的实发工资*/
 case 4:
 case 5:
 case 6:
      salary=salary+sales*0.07; /*3000 元<销售额≤6000 元时的实发工资*/
      break;
 case 7:
 case 8:
 case 9:
 case 10:
      salary+=sales*0.10;          /*6000 元<销售额≤10 000 元时的实发工资*/
      break;
 default:
      salary=salary+sales*0.13;      /*销售额>10 000 元时的实发工资*/
      break;
}
printf("该员工实发工资为: %.2f 元\n\n", salary); /*输出员工的实发工资*/
}
}

```

以上程序当输入的销售额为负数时就会出错, 运行结果如图 5-12 所示。



```

请输入一个员工月销售额: 5800
该员工实发工资为: 1406.00元

请输入一个员工月销售额: -12500
该员工实发工资为: -219.00元

请输入一个员工月销售额:

```

图 5-12 例 5.36 的运行结果

2. 等价类划分法和边界值分析法

等价类划分法是黑盒法中一种非常实用的测试方法, 它针对具体程序的输入数据, 划分为若干有效等价类和无效等价类测试某等价类的代表值等价于对这一类中其他值的测试。

由于许多错误往往发生在输入输出的边界上, 所以边界值分析法是对等价类划分法的补充, 这种方法不是取等价类中的任意值, 而是选取恰好等于、稍大于或稍小于边界值的值作为测试数据。表 5-2 给出了详细的测试用例, 让这两种方法珠联璧合, 可迅速、全面地找出程序中的逻辑错误。

例 5.37 以下是错误的程序。某商店销售某种记录本的单价与购买量关系如下, 要求计算顾客应付金额。

单价（元）	购买量（本）
3.6	$x \geq 10$
4.5	$3 \leq x < 10$
5.2	$x < 3$

```
#include <stdio.h>
main()
{int x;
 float y;
 printf("请输入顾客的购买量 ");
 scanf("%d" , &x);
 if (x>10) y=3.6*x;
 else if (3<=x<10) y=4.5*x;
 else y=5.2*x;
 printf("购买量 %d 金额 %.2f\n" , x,y);
}
```

表 5-2 例 5.37 测试用例

输入数据	预期正确结果	实际运行结果	发现错误	理论依据
2	10.40	9.00	错误: if (3<=x<10) y=4.5*x; 改正: if (x>=3&& x<10) y=4.5*x; 或 if (x>=3) y=4.5*x;	等价类划分法
3	13.50	13.50		边界值分析法
5	22.50	22.50		等价类划分法
10	36.00	45.00	错误: if (x>10) y=3.6*x; 改正: if (x>=10) y=3.6*x;	边界值分析法
20	72.00	72.00		等价类划分法

例 5.38 以下是错误的程序。编程从键盘输入 10 个学生的高考成绩，求高考最高分和最低分。

```
#include <stdio.h>
#define N 10 /*N 符号常量表示学生人数*/
main()
{int i , a[N] , max, min ;
 for (i=0; i<N; i++)
 {printf("请输入第%d 个同学的分数", i+1 );
 scanf("%d" , &a[i]); } /*输入 10 学生的高考成绩存放到数组 a 中*/
max=-1; min=10000;
 for (i=0; i<N; i++)
 {if (a[i]>max) max=a[i]; /*max 存放最高分*/
 else if (a[i]<min) min=a[i]; /*min 存放最低分*/
 }
 printf("高考最高分: %d 高考最低分: %d \n " ,max , min);
}
```

两次运行以上程序的结果如图 5-13、图 5-14 所示。从运行结果可以看出第 1 次运行输入的 10 个高考分是无序的，未发现程序运行结果有错误。第 2 次运行输入的 10 个高考分是由低到高有序的，可发现程序运行结果有错误。

正确的程序见第2章例2.28。

```
请输入第1个同学的分数603
请输入第2个同学的分数591
请输入第3个同学的分数587
请输入第4个同学的分数612
请输入第5个同学的分数623
请输入第6个同学的分数587
请输入第7个同学的分数572
请输入第8个同学的分数575
请输入第9个同学的分数601
请输入第10个同学的分数588
高考最高分: 623  高考最低分: 572
```

图 5-13 例 5.38 第 1 次运行结果

```
请输入第1个同学的分数575
请输入第2个同学的分数578
请输入第3个同学的分数580
请输入第4个同学的分数583
请输入第5个同学的分数586
请输入第6个同学的分数592
请输入第7个同学的分数595
请输入第8个同学的分数597
请输入第9个同学的分数603
请输入第10个同学的分数622
高考最高分: 622  高考最低分: 10000
```

图 5-14 例 5.38 第 2 次运行结果

3. 语句覆盖法

语句覆盖是白盒法的一种。实际上就是设计若干个测试用例运行被测程序使每条语句至少执行一次，以下例程有逻辑错误。

例 5.39 以下是错误的程序。函数 fun 的功能是：根据输入的三条边长，判断其能否构成三角形及构成三角形的类型。

若能构成等边三角形，函数返回值为 3；

若能构成等腰三角形，函数返回值为 2；

若能构成三角形，函数返回值为 1；

若不能构成三角形，函数返回值为 0。

要求：改正 fun 函数中的错误，不能增、删语句，不能更改程序的结构。

```
#include <stdio.h>
int fun(int a ,int b, int c)
{if(a+b>c&& b+c>a&&a+c>b)
    {if (a==b&&b==c)
        return 1;
        else if(a==b||b==c||a==c)
            return 2;
        else return 3;
    }
    else return 0;
}

main()
```

```
{int a,b,c,shape;
printf("input a b c");
scanf("%d%d%d" , &a,&b,&c);
printf("a=%d, b=%d, c=%d\n" ,a,b,c);
shape=fun(a,b,c);
printf("\n\nthe shape: %d\n" ,shape);
}
```

例 5.39 测试用例见表 5-3。

表 5-3 例 5.39 测试用例

输入数据	预期正确结果	实际运行结果	发现错误	理论依据
2 3 7	0	0		语句 覆盖法
3 4 5	1	3	错误: return 3; (第 8 行) 改正: return 1;	
3 5 3	2	2		
6 6 6	3	1	错误: return 1; (第 5 行) 改正: return 3;	

4. 回溯法和安插显示语句法

这两种方法均属于软件调试方法。回溯法的基本思想是根据程序中出错的“征兆”，人工沿程序的控制流程，往回追踪源程序代码，直到找出错误根源或产生错误的范围。但由于有时程序的出错范围或“征兆”难以确定，因此需与后一种方法配合使用，才能使隐蔽的错误显现出来。现将一例程抽象成如下形式：

```
while (循环表达式)
{...../*顺序执行的语句*/
  if (分支表达式)
    printf("111\n");      /*安插的显示语句*/
    语句 1;
  else
    printf("222\n");      /*安插的显示语句*/
    语句 2;
  ...../*顺序执行的语句*/
}
```

从程序执行结果判断程序本应执行而实际没执行语句 1，于是应首先怀疑 if 语句的表达式写错了，所以在语句 1 之前插入一条 printf(“111\n”)语句，这样如果程序走这个分支，执行到这句一定会显示 111，结果却发现程序不执行这个分支。因此又推测可能走了另一个分支，于是在语句 2 之前插入 printf(“222\n”)语句。再执行程序，发现也没执行这个分支。而在 if 语句前都是顺序执行的语句，再无分支了，于是判定循环一次也没执行。因此问题出在 while(循环表达式) 语句上，而循环表达式是!feof(fp)，即文件位置指针未指向文件尾时才能循环，所以最后“确诊”在循环之前一定有语句把指针移向了文件尾，再向上追踪果然不出所料，因此，在循环语句之前插入 rewind(fp)，问题迎刃而解。

例 5.40 以下是错误程序。输出 100~300 之间的素数，要求每行输出 10 个素数。

```
#include <stdio.h>
#include <math.h>
```

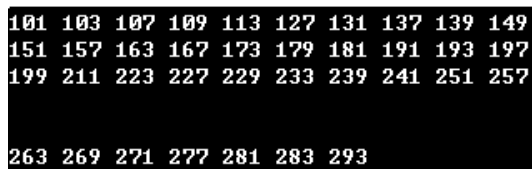


```

main()
{int  n ,k , i, m=0;
  for(n=101; n<=300; n=n+2)
  {k=sqrt(n);
   for(i=2; i<=k; i++)
    if (n%i==0) break;
   if (i>=k+1)
    {printf("%d ",n);    /*符合 i>=k+1  输出该素数*/
     m=m+1;              /*统计素数的个数*/
    }
   if (m%10==0) printf("\n");
  }
  printf("\n");
}

```

从图 5-15 的运行结果可以看出在输出素数 257 之后, 当 n 值为 259、261 这两个非素数时, 也执行了 `if (m%10==0) printf("\n");` 语句换行, 因此, 导致输出结果多出两个空行与原意不符。由此发现程序的控制流程有错误, 需要将 `if (m%10==0) printf("\n");` 语句移动到 `if (i>=k+1)` 语句的大括号内。修改后的运行结果如图 5-16 所示。



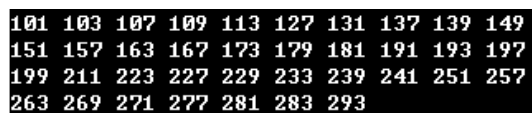
```

101 103 107 109 113 127 131 137 139 149
151 157 163 167 173 179 181 191 193 197
199 211 223 227 229 233 239 241 251 257

263 269 271 277 281 283 293

```

图 5-15 例 5.40 的运行结果



```

101 103 107 109 113 127 131 137 139 149
151 157 163 167 173 179 181 191 193 197
199 211 223 227 229 233 239 241 251 257
263 269 271 277 281 283 293

```

图 5-16 修改例 5.40 程序之后的运行结果

5. 归纳法

归纳法也是软件调试方法之一, 它的基本思想是从一些线索(错误现象)着手, 通过分析它们之间的关系找出错误。

例 5.41 以下是错误程序。编程求解“百元买百鸡”问题。

已知公鸡每只 5 元, 母鸡每只 3 元, 小鸡 3 只 1 元。用 100 元买 100 只鸡, 问公鸡、母鸡、小鸡各多少只?

```

#include <stdio.h>
main()
{int  x, y, z;
  printf("%8s%8s%8s\n", "公鸡", "母鸡", "小鸡");
  for (x=1; x<=19; x++)
    for (y=1; y<=33; y++)
      {z=100-x-y;

```

```
    if (5*x+3*y+z/3==100)
        printf("%8d%8d%8d\n", x, y, z);
}
```

根据原意，小鸡数应是 3 的倍数，从图 5-17 所示的运行结果来看，小鸡数不是 3 的倍数的那 3 行输出结果都是错误的。因此需要将 `if(5*x+3*y+z/3==100)` 修改为：
`if(5*x+3*y+z/3.0==100)` 得到正确的运行结果，如图 5-18 所示。

公鸡	母鸡	小鸡
3	20	77
4	18	78
7	13	80
8	11	81
11	6	83
12	4	84

图 5-17 例 5.41 的运行结果

公鸡	母鸡	小鸡
4	18	78
8	11	81
12	4	84

图 5-18 例 5.41 程序修改后正确的运行结果

程序调试是编程与测试过程中的重要环节与手段，至今尚无完善的方法。只有在调试实践中不断探索、不断总结，才能逐渐应对程序中可能出现的各种问题，提高调试程序的能力。

5.3 C 程序的调试技巧

1. 设置断点单步执行

(1) 变量未赋值

类似于例 5.13 求 1~100 之间偶数的累加和程序，初学者对于运行结果为负数会感到匪夷所思，对于这类存放累加和的变量未赋初值的错误，在求和的语句处设置断点并单步执行，观察变量的值，问题就会变得明朗了。

图 5-19 是 VC++ 6.0 工具栏中的两个命令按钮，功能如下：

左边的为开始调试按钮，对应快捷键 F5。

右边的小手图标是设置断点或取消断点的切换按钮，对应快捷键 F9。在调试之前先编译源程序，使两个按钮转变为可用状态。

单击图 5-20 所示的 `sum=sum+i;` 语句行，将光标定位在该行→单击设置断点的命令按钮→单击开始调试按钮，屏幕下方显示出 `sum=sum+i;` 语句执行前 `sum` 的值是负数，如图 5-21 所示，这显然与原意不符。因此按 F10 键单击执行命令，如图 5-22 所示是执行一次 `sum=sum+i;` 语句之后各变量的值，发现 `sum` 的负数值是与第一个偶数 2 相加的结果。



图 5-19 调试、设置断点命令按钮

```
#include <stdio.h>
main( )
{int i, sum;
  for(i=2; i<=100; i+=2)
    sum=sum+i;
  printf("sum=%d\n", sum);
}
```

图 5-20 设置源程序中的断点

名称	值
i	2
sum	-858993460
Auto Locals this	

图 5-21 sum=sum+i;语句执行前的变量值

名称	值
i	2
sum	-858993458
Auto Locals this	

图 5-22 sum=sum+i;语句执行后的变量值

(2) 字符数组定义的长度不够

例 5.42 以下是错误程序。实现给两个字符串赋值并输出。

以上程序并没有如预期的那样，第 1 行输出单词字符串，第 2 行输出数字字符串。运行结果如图 5-24 所示。

如图 5-23 所示，程序设置断点在 strcpy(s2,"123456");语句处，单击开始调试按钮，屏幕下方显示出断点语句执行前的字符数组值，如图 5-25 所示。

```
#include <stdio.h>
#include <string.h>
main( )
{char s1[10];
 char s2[4];
 strcpy(s1,"computer");
 ● strcpy(s2,"123456");
 printf("%s\n",s1);
 printf("%s\n",s2);
}
```

56
123456

图 5-23 设置了断点的程序

图 5-24 例 5.42 的运行结果

名称	值
s1	0x0012ff3c "computer"
s2	0x0012ff38 "烫烫computer"
s2[4]	99 'c'
Auto Locals this	

图 5-25 断点语句执行前的字符数组值

按 F10 键单步执行断点语句后，字符数组值如图 5-26 所示。可以看出给 s2 赋值超过所容纳的 4 个字节时就开始覆盖 s1 的内容了，因此出现了出乎预料的运行结果。

名称	值
s1	0x0012ff3c "56"
s2	0x0012ff38 "123456"
strcpy returned	<void>
Auto Locals this	

图 5-26 断点语句执行后的字符数组值

大多数 C 程序集成开发环境中的调试都提供了这三种功能：单步运行（Step over）、跟踪（Step into）和跳出（Step out）。

单步运行命令是控制程序一次执行一行（不管这一行中是否有函数调用）。

跟踪命令是控制程序沿流程向前走一步，如果当前要执行的语句中有函数调用时，就进入被调用的函数里面去执行；如果当前要执行的语句中没有函数调用时，那就执行完该语句，这时的效果与单步运行是一样的。

跳出命令通常与跟踪命令配合使用，作用是执行完当前所在函数剩余的语句并跳出该函数。

VC++ 6.0 调试程序常用快捷键如下。

命令	快捷键	功能
Insert/ Remove Breakpoint	F9	设置断点或取消断点
G0	F5	开始调试，运行程序至断点
Stop Debug	Shift+F5	停止调试
Step Over	F10	单步执行，不进入函数
Step Into	F11	单步执行，进入被调函数
Step Out	Shift+F11	跳出当前函数，返回调用处
Run to cursor	Ctrl+F10	运行至光标处
Break		从当前位置退出，终止程序运行

2. 建立和运行包含多个文件的 C 程序

一个复杂的 C 程序可以划分为主函数和多个自定义函数，分别存放在不同的文件中。要建立和运行包含多个文件的 C 程序，需要建立工程文件和它的工作区。比较简单的方法是用户只建立工程文件，而工作区由系统自动建立。建立工程文件后，依次单击“工程”→“添加到工程”→“文件”选项把多个源程序文件添加到工程中，然后编译、链接、运行程序。

附录 A

C 语言网络资源分布简介

1. C 语言网

<http://www.cyuyan.com.cn/>

栏目：C 语言论坛、C++论坛、技术问答、代码分享等。

2. C 语言中文网

<http://c.biancheng.net/cpp/>

栏目：C 语言教程、C 语言函数手册、C 语言代码、C 语言文章等。

3. CSDN

<http://www.csdn.net/>

栏目：C 语言教程、C 语言书籍、资料等。

4. 网易云课堂

<http://study.163.com/>

栏目：大量 C 语言视频

5. 百度传课

<http://www.chuanke.com/>

栏目：大量 C 语言视频

附录 B

实验报告格式要求

C 语言程序设计实验报告

班级	学号	姓名	成绩
实验 X	XXXXXX		

例：
实验五 循环程序设计

一、实验目的和要求

二、实验项目和程序清单

1. 上机实验 1 PXX
程序清单

例：上机实验 1 P48
教材第 48 页第 1 题

运行结果

注明该程序是否调试
通过

运行结果截图

三、实验思考和总结

思考题及经验总结：
解答教材中的思考题
或总结自己的编程调
试经验、教训

参 考 文 献

- [1] 谭浩强. C 程序设计（第四版）. 北京：清华大学出版社，2010.
- [2] 谭浩强. C 程序设计（第四版）学习辅导. 北京：清华大学出版社，2010.
- [3] 杜忠友等. C 语言及程序设计. 北京：中国铁道出版社，2008.
- [4] 王成端等. C 语言程序设计实训. 北京：中国水利水电出版社，2005.
- [5] 常琳等. C 语言程序设计习题解答与实验指导. 北京：北京大学出版社，2010.
- [6] 吴启武等. C 语言课程设计案例精编（第 3 版）. 北京：清华大学出版社，2016.
- [7] 李峰等. C 语言程序设计上机指导与习题选解. 上海：复旦大学出版社，2011.
- [8] 曹化工等. C 语言程序设计典型题解析及自测试题（第 2 版）. 西安：西北工业大学出版社，2008.
- [9] 全国计算机等级考试命题研究组. 2014 年全国计算机等级考试超级题库真题库+样题库二级 C 语言程序设计库（第 4 版）. 北京：北京邮电大学出版社，2014.
- [10] [美]K.N.King. 吕秀锋等译. C 语言程序设计：现代方法（第 2 版）. 北京：人民邮电出版社，2010.
- [11] 赵池龙等. 实用软件工程（第 4 版）. 北京：电子工业出版社，2015.
- [12] 周元哲. 软件测试教程. 北京：机械工业出版社，2010.